ePOS-Device SDK for iOS ユーザーズマニュアル

概要

特徴および動作環境について説明します。

環境の構築

ePOS-Device Service I/Fの環境構築方法について説明します。

プログラミングガイド

アプリケーション開発のプログラミング方法について説明します。

API リファレンス

ePOS-Device SDK for iOSで提供しているAPIについて説明します。

サンプルプログラム

サンプルプログラムについて説明します。

付録

TM-DTシリーズで使用する場合

M00078002
Rev.C

ご注意

- 本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固くお断りします。
- 本書の内容については、予告なしに変更することがあります。最新の情報はお問い合わせください。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がありましたらご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。
- 本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、またはエプソンおよびエプソン指定の者以外の第三者により修理・変更されたことなどに起因して生じた損害などにつきましては、責任を負いかねますのでご了承ください。
- エプソン純正品およびエプソン品質認定品以外のオプションまたは消耗品を装着してトラブルが発生した場合には、責任を負いかねますのでご了承ください。

商標について

EPSON、EXCEED YOUR VISION、ESC/POS は、セイコーエプソン株式会社の登録商標です。

iPhone®、iPod touch®、iPad®、iTunes®、Xcode® は、米国および他の国々で登録された Apple Inc. の商標です。

iOS® は、Cisco の米国およびその他の国における商標または登録商標です。

Wi-Fi® は、Wi-Fi Alliance® の登録商標です。

ESC/POS® コマンドシステム

EPSON は、独自の POS プリンターコマンドシステム、ESC/POS により、業界のイニシアチブをとってきました。ESC/POS は特許取得済みのものを含む数多くの独自のコマンドを持ち、高い拡張性で多才な POS システムの構築を実現します。ほとんどの EPSON POS プリンターとディスプレイに互換性を持つ他、この独自の制御システムにはフレキシビリティーもあるため、将来アップグレードが行いやすくなります。その機能と利便性は世界中で評価されています。

安全のために

記号の意味

本書では以下の記号が使われています。それぞれの記号の意味をよく理解してから製品を取り扱ってください。

注意	ご使用上、必ずお守りいただきたいことを記載しています。この表示を無視して誤った取り扱いをすると、製品の故障や動作不良の原因になる可能性があります。
参考	補足説明や知っておいていただきたいことを記載しています。

使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度などにおいて高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全維持のためにフェールセーフ設計や冗長設計の措置を講じるなど、システム全体の安全設計にご配慮いただいた上で弊社製品をご使用いただくようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、きわめて高い信頼性・安全性が必要とされる用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認の上、ご判断ください。

本書について

本書の目的

本書は、TM-i を使った、ePOS-Device のシステムの構築、アプリケーションの開発、設計に必要なすべての情報を開発技術者に提供することを、その目的としています。

本書における TM-i シリーズとは、以下のプリンターの総称です。

- TM-T70-i
- TM-T88V-i

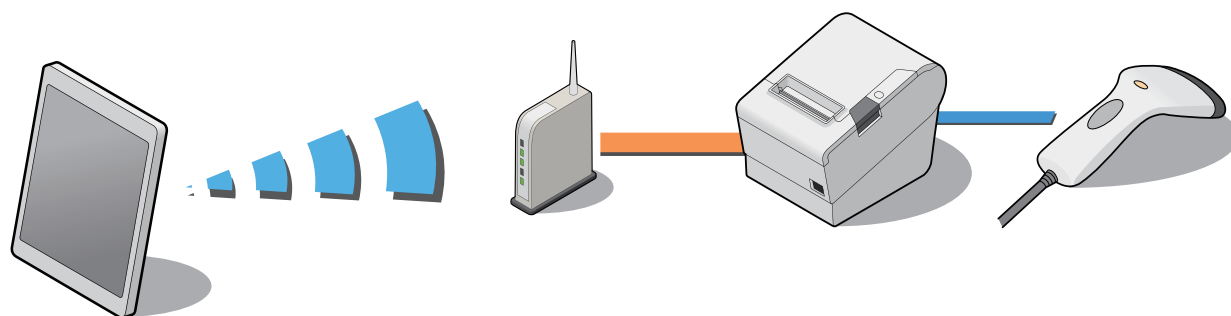
本書の構成

本書は次のように構成されています。

もくじ

概要

本章では、ePOS-Device SDK for iOS の特徴および仕様について説明しています。

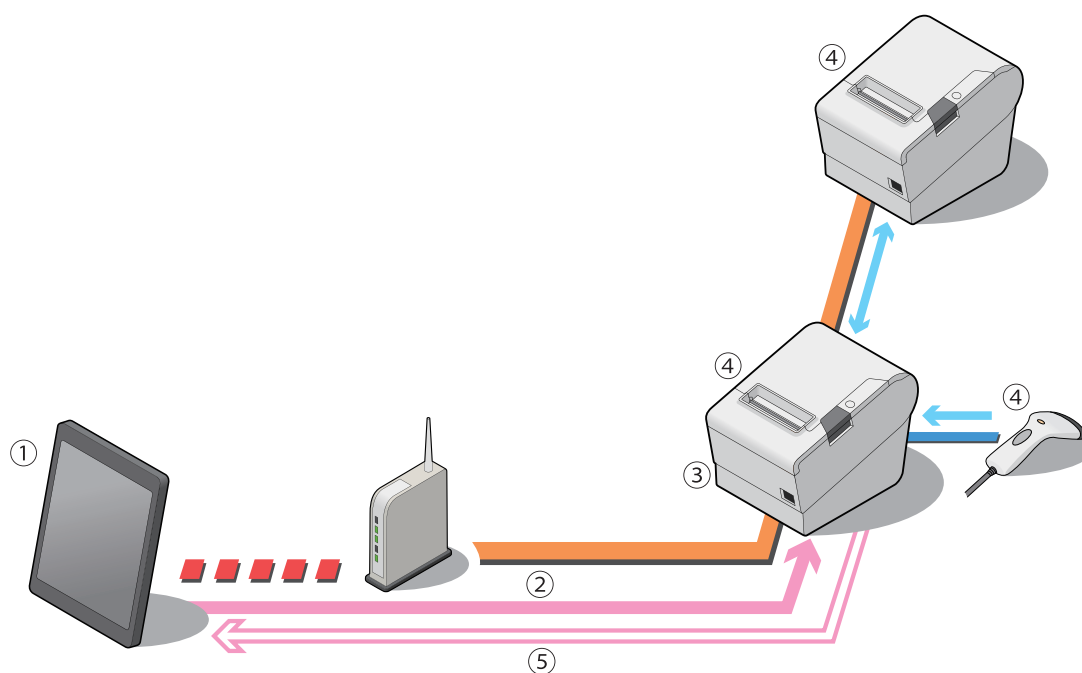
ePOS-Device SDK

ePOS-Device SDK for iOS は、TM-i に接続された各種周辺機器を制御するための iOS アプリケーションを開発する、開発者向け SDK です。ePOS-Device SDK で提供する API を使用してアプリケーションを開発します。

ePOS-Device SDK には、Android アプリケーション向けの ePOS-Device SDK for Android も用意されています。

特長

- ネットワーク接続可能な環境であれば、どこからでも使えます。
- 周辺機器の制御用に別途、コントローラーとなる機器 (コンピューターなど) を用意する必要がありません。
- ePOS-Device SDK を使ってデバイスにアクセスすると、デバイスは自動で排他ロックされます。
  複数端末から同時アクセスしても、制御が混在しません。デバイス制御していた端末がデバイスを解放すると、他の端末から制御可能となります。
- アプリケーション同士で、データを送受信できます。
  詳細は 9 ページ「コミュニケーションボックスの使い方」を参照してください。
- 印刷ジョブ ID を指定してプリンターに印刷処理をリクエストできます。*
  詳細は 12 ページ「アプリケーションから印刷ジョブ ID を指定」を参照してください。
- プリンターの印刷終了の応答を待たずに、印刷データをプリンターに送信できます。(スプーラー)*
  詳細は 13 ページ「スプーラー機能」を参照してください。
- プリンターがエラー応答したり (用紙なし、カバーオープンなど)、レスポンスを返さなかったりした場合、別のプリンターに印刷データを送信できます。(迂回印刷)*
  詳細は 14 ページ「迂回処理」を参照してください。

* TM-i ファームウェア Ver.4.1 以降対応

システム構成例

1 iOS デバイスに iOS アプリケーションを配置します。

2 iOS アプリケーションから TM-i にリクエストメッセージを送信します。

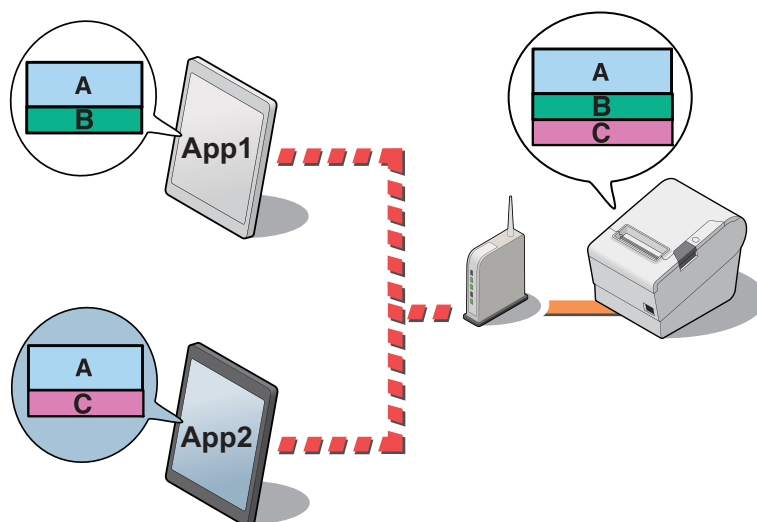
3 TM-i から制御可能なデバイスにデータを送信します。

4 TM-i に接続されているデバイスやネットワークプリンターを制御します。

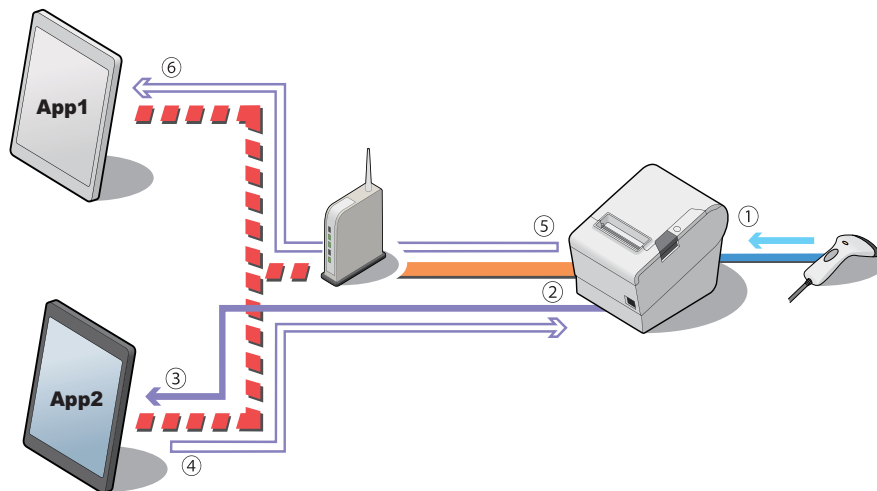
5 TM-i からアプリケーションにレスポンスが返されます。

コミュニケーションボックスの使い方

コミュニケーションボックスとは、アプリケーション間でデータを送受信したりするための、ePOS-Device Service I/F が提供するデータの仮想空間です。

注意	ePOS-Device が提供しているのは、アプリケーション間でデータを送受信するための仮想空間と仕組みです。送受信するデータは、お客様のアプリケーションの仕様に従って作成してください。

コミュニケーションボックスには、以下の活用方法があります。

- iOS デバイスを POS の表示端末に使用
- デジタルサイネージを表示
- 受付入力

コミュニケーションボックスの使用したデータ処理例

1 TM-i がスキャナーで読み取ったバーコードデータを受信します。

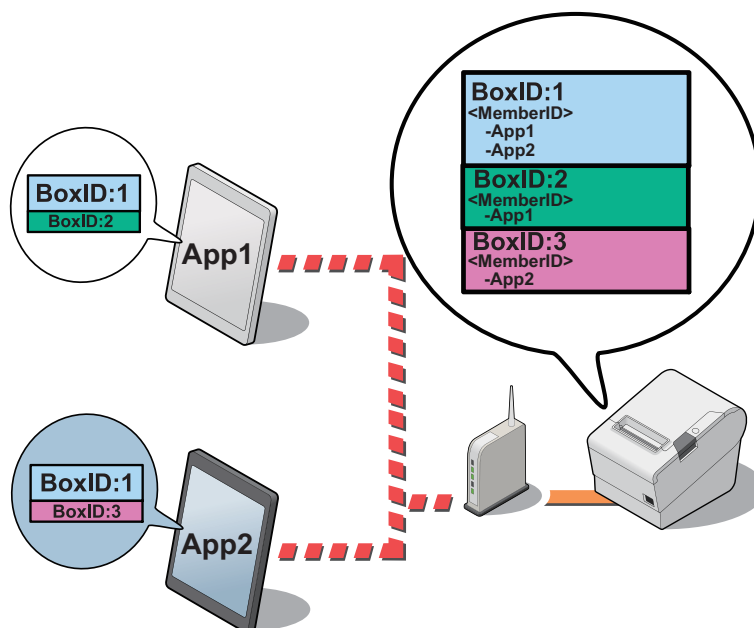
2 ePOS-Device Service I/F は、バーコードデータを App2 に通知します。

3 App2 はバーコードデータを取得し、POS データに変換します。

4 App2 は、ePOS-Device Service I/F のコミュニケーションボックスに表示データを送信します。

5 ePOS-Device Service I/F は、App1 にコミュニケーションボックスに表示データが格納されたことを通知します。

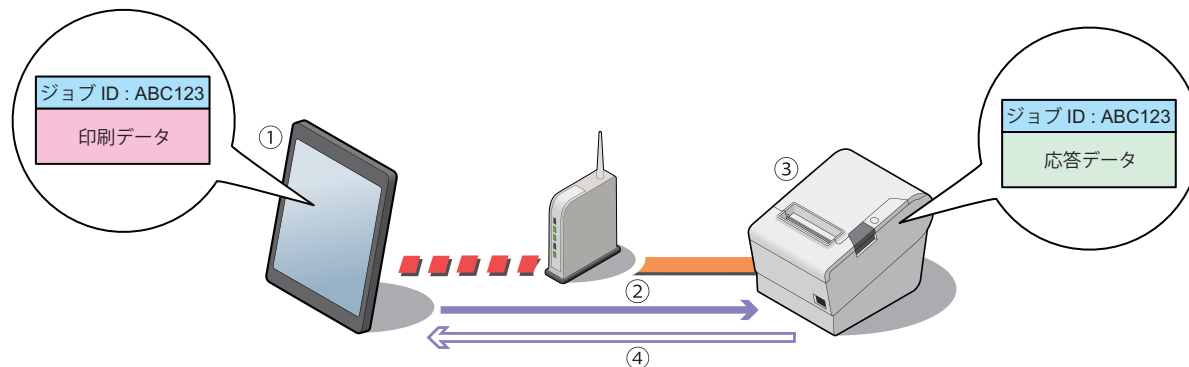
6 App1 は、コミュニケーションボックスに格納された表示データを取得します。

コミュニケーションボックスの仕組み

ePOS-Device Service I/F は、コミュニケーションボックスをボックス ID(上図 :BoxID) で管理します。
コミュニケーションボックスに所属しているアプリケーション同士でデータの送受信が可能になります。
上図の場合、App1 と App2 のアプリケーションは、BoxID:1 のコミュニケーションボックスを使って、アプリケーション間でデータの送受信ができることになります。

コミュニケーションボックスの仕様

作成できるコミュニケーションボックスの最大数	20 個
1 つのコミュニケーションボックスに所属できるアプリケーションの最大数	20 個
1 つのコミュニケーションボックスが保持できる送信履歴の容量	10240 Byte
1 度に送信できるデータのサイズ	1024 Byte

アプリケーションから印刷ジョブ ID を指定

アプリケーションから印刷ジョブ ID を指定してリクエストを送信すると、指定した印刷ジョブ ID を持つレスポンスを返します。(TM-i ファームウェア Ver.4.1 以降)

スプーラー機能

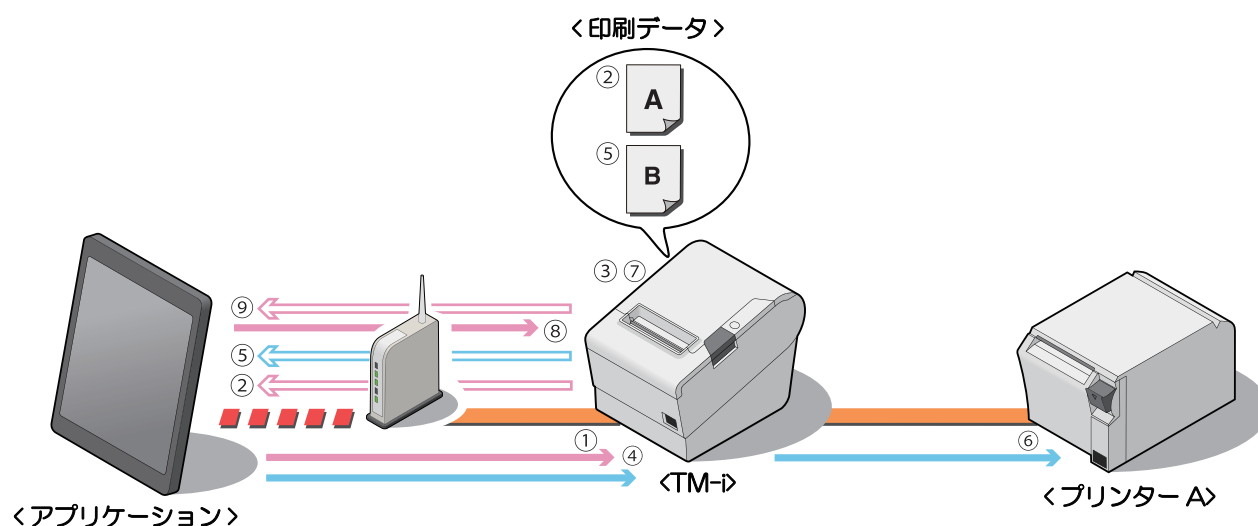
スプーラー機能とは、アプリケーションから受け取った印刷データを TM-i に一時的に蓄えておき、順次バックグラウンド印刷する機能です。(TM-i ファームウェア Ver.4.1 以降)

TM-i は、印刷データをスプーラーに入れた時点でアプリケーションに応答データを返すので、アプリケーションは印刷完了を待たずに次の処理に移ることができます。

アプリケーションは、応答データに含まれるジョブ ID を使って印刷状況を問い合わせることもできます。

スプーラーは、EPSON TMNet WebConfig で設定します。
以下の設定をします。詳細は、各 TM-i の詳細取扱説明書を参照してください。

- スプーラーの有効設定
- 印刷処理のリトライ設定

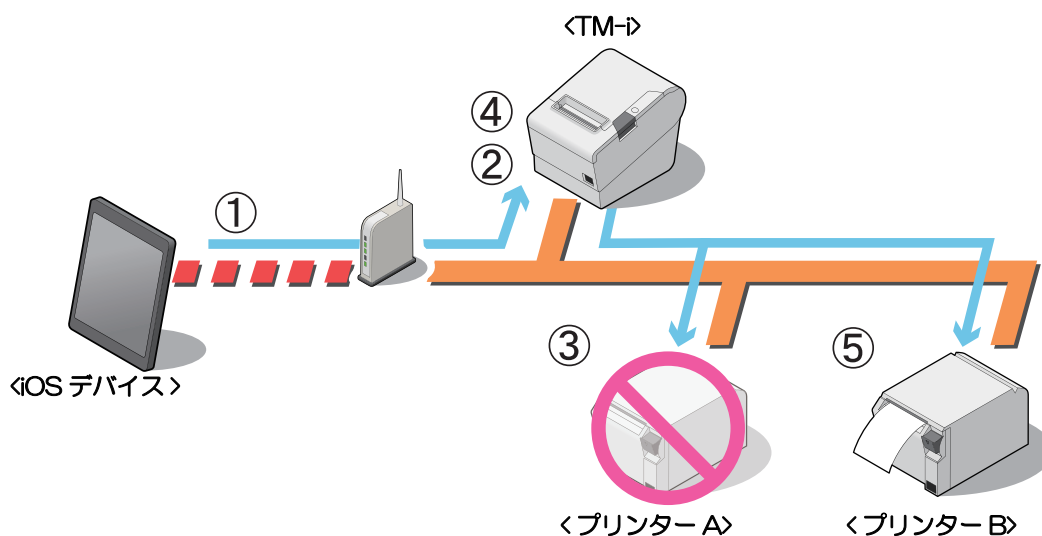
プリンターがリトライ処理で無応答の場合、迂回処理で別のプリンターに印刷することができます。
14 ページ「迂回処理」を参照してください。

1. iOS デバイスから、印刷データ A を TM-i(ePOS-Device Service I/F) に送信します。
2. TM-i はスプーラーに印刷データ A を保存し、iOS デバイスに応答データを返します。
3. TM-i はスプーラーから印刷データ A を取り出し、TM-i に印刷命令を出します。
4. iOS デバイスから、印刷データ B を TM-i に送信します。
5. TM-i はスプーラーに印刷データ B を保存し、iOS デバイスに応答データを返します。
6. TM-i はスプーラーから印刷データ B を取り出し、プリンター A に印刷命令を出します。

7 印刷データ A が完了します。

8 iOS デバイスから、印刷データ A の結果を TM-i に問い合わせます。

9 TM-iは印刷データAが正常に終了したことを応答データとしてiOSデバイスに返します。

迂回処理

> **参考** 迂回処理の設定は、EPSON TMNet WebConfig で設定します。詳細は、各 TM-i の詳細取扱説明書を参照してください。

1 iOS デバイスから、印刷データを TM-i(ePOS-Device Service I/F) に送信します。

2 TM-i は、プリンター A に印刷命令を出します。

3 プリンター A は、応答しません。

4 TM-i は、プリンター A に印刷命令を再試行します。
プリンター A が再試行に応答しなかった場合、プリンター B に印刷命令を出します。

5 iOS デバイスから送信した印刷データが、プリンター B に印刷されます。

動作環境

iOS バージョン

- ❏ iOS Version 4.2 ～ 4.3.5
- ❏ iOS Version 5.0 ～ 5.1.1
- ❏ iOS Version 6.0 ～ 6.1.4
- ❏ iOS Version 7.0 ～ 7.1.2
- ❏ iOS Version 8.0 ～ 8.0.2

参考	最新のバージョンは、README ファイルを参照してください。

iOS デバイス

- ❏ iPhone 4/ iPhone 4s/ iPhone 5/ iPhone 5s/ iPhone 5c/ iPhone 6/ iPhone 6 Plus
- ❏ iPod touch (3rd generation)/ iPod touch (4th generation)/ iPod touch (5th generation)
- ❏ iPad/ iPad 2/ iPad (3rd generation)/ iPad (4th generation)/ iPad Air
- ❏ iPad mini/ iPad mini with Retina display

TM-i

- ❏ TM-T88V-i (TM-i ファームウェア Ver.4.0 以降)
- ❏ TM-T70-i (TM-i ファームウェア Ver.4.0 以降)

TM プリンター

TM-i のプリンターとは別に、以下の TM プリンターを制御できます。

TM プリンター	TM-i ファームウェア	
	Ver.4.0	Ver.4.1
TM-L90*	○	○
TM-P20	-	○
TM-P60II	○	○
TM-T70*	○	○
TM-T70II*	○	○
TM-T88IV*	○	○
TM-T88V*	○	○
TM-T90*	○	○
TM-T90II*	○	○
TM-T90KP*	○	○

* イーサネットモデル (UB-E02/ UB-E03)、または無線 LAN モデル (UB-R03/ UB-R04) 。
そのほかのインターフェイスのモデルは使用できません。

インターフェイス

以下のインターフェイスの TM プリンターをお使いください。

❏ イーサネット

❏ 無線 LAN (UB-R03/ UB-R04)

サポート周辺機器

❏ カスタマーディスプレイ
- DM-D110 (USB 仕様)

❏ 標準 HID デバイス
- キーボードデバイス
- バーコードスキャナー

❏ シリアル通信デバイス

参考　本 SDK でシリアル通信デバイスを制御する場合、コマンドを送受信する API を使用して制御します。

開発環境

iOS アプリケーションを開発するには、以下が必要です。

- Xcode Version 4.2 ～ 4.6
- Xcode Version 5.0 以降

提供物

パッケージ

ファイル名	説明
ePOS-Device.h	API 用 Objective-C ヘッダーファイルです。
libeposdevice.a	機能実行用ライブラリーです。 (i386、x86_64、ARMv7、ARMv7s、arm64 に対応)
ePOS-Device_Sample_iOS.zip	サンプルプログラムファイルです。
README.jp.txt	README ファイルです。
EULA.jp.txt	SOFTWARE LICENSE AGREEMENT が記載されています。
ePOS-Device_SDK_iOS_ja_revx.pdf	本書です。ePOS-Device SDK for iOS のプログラミング方法や API の説明をしています。

ダウンロード

提供物は、下記エプソン販売ホームページからダウンロードできます。

http://www.epson.jp/support/sd/

制限事項

- 罫線コマンドは、プリンターが「罫線コマンド」をサポートしていないとお使いいただけません。
- ドロアーとブザーは一緒に使用できません。
- ブザーの機能は、プリンターにブザーが付いてなければ使用できません。
- 2 次元バーコードリーダーを使用する場合、日本語などのマルチバイト文字は正しく取得できません。
- 2 次元バーコードのデータに ASCII 制御コード (0x00 ～ 0x1F) が含まれる場合、制御コードは取得できません。
- キーボードから取得できるキーコードに制限があります。取得可能なキーコードは 147 ページ「キーコード一覧」を参照してください。

環境の構築

本章では、ePOS-Device Service I/F の環境構築について説明しています。

作業フロー

1. ネットワークの設定

システムおよび TM-i のネットワーク設定をします。
以下の方法で設定できます。詳細は、各 TM-i の詳細取扱説明書を参照してください。

- EPSON TMNet WebConfig
- EpsonNet Config
- 簡単キッティング

2. TM-i にデバイスを接続

TM-i にデバイスを接続します。接続可能なデバイスは、15 ページ「動作環境」を参照してください。

3. デバイスの登録 (20 ページ)

接続するデバイスを TM-i に登録します。
登録は、EPSON TMNet WebConfig を使って行います。
詳細は、各 TM-i の詳細取扱説明書を参照してください。

デバイスの登録

ePOS-Device SDK for iOS が制御するデバイスを、TM-i に登録します。
EPSON TMNet WebConfig を使って設定します。EPSON TMNet WebConfig は、Web ブラウザーから起動します。詳細は、各 TM-i の詳細取扱説明書を参照してください。

TM-i に接続するデバイスごと、以下を設定します。

プリンター

項目	説明
デバイス ID	制御するプリンターの ID を入力します。(任意の文字列)
型番	制御するプリンターの型番を選択します。
IP アドレス	デバイス ID ごとのプリンターの IP アドレスを設定します。
リトライ間隔	タイムアウトのリトライ間隔を設定します。

カスタマーディスプレイ

項目	説明
カスタマーディスプレイ	使用の有無を設定します。
通信設定	通信速度、データビット、パリティーを設定します。
輝度	カスタマーディスプレイの輝度を設定します。

参考

カスタマーディスプレイのデバイス ID は、"local_display" 固定です。

キー入力デバイス

項目	説明
デバイス ID	デバイス ID を入力します。(任意の文字列)
デバイス名	キー入力デバイスのデバイス名を選択します。
制御スクリプト	キー入力デバイスで使用するデバイス制御スクリプトを選択します。

シリアル通信デバイス

項目	説明
シリアル通信デバイス	使用の有無を設定します。
通信速度	デバイスの通信速度を設定します。
データビット	データビットを設定します。
パリティ	パリティーを設定します。
ストップビット	ストップビットを設定します。
フロー制御	フロー制御を設定します。

参考

シリアル通信デバイスのデバイス ID は、"local_serial" 固定です。

プログラミングガイド

本章では、ePOS-Device SDK for iOS を使用したアプリケーション開発のプログラミング方法について説明します。

ePOS-Device SDK for iOS の組み込み方法

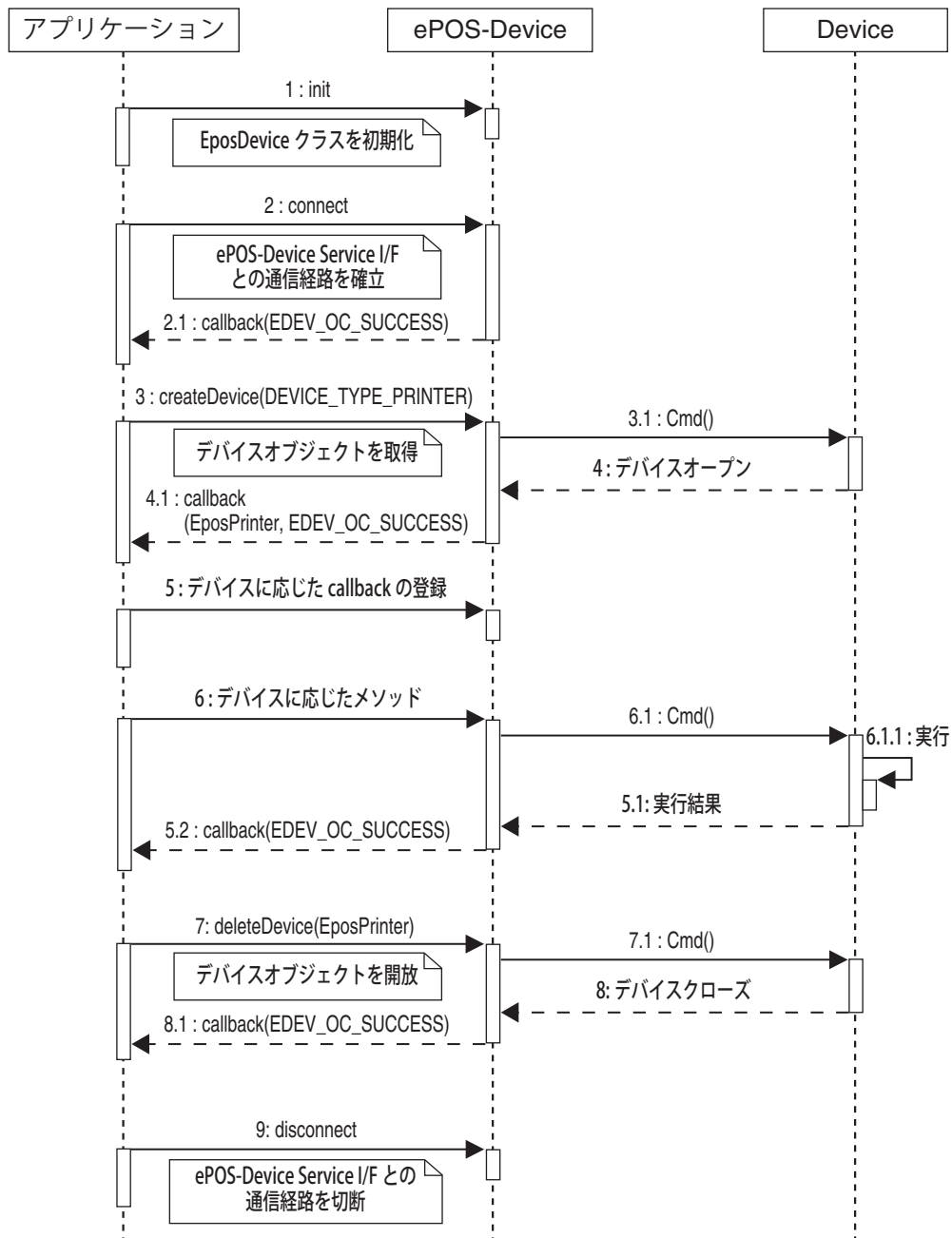
ePOS-Device SDK for iOS の組み込み方法について説明します。
以下の手順で組み込んでください。

1. Xcode で新しいプロジェクトを作成します。
2. 提供されたObjective-Cヘッダー(ePOS-Device.h)をXcode の[Project Navigator]の対象プロジェクトの任意の階層にドラッグします。
3. 提供されたスタティックライブラリー (libeposdevice.a) を Xcode の [Project Navigator] の対象プロジェクトの任意の階層にドラッグします。
4. [Project Navigator] から、ルートにある Project ファイルを選択します。
5. [Targets] の [Build Phase] を選択します。
6. [Link Binary With Libraries] を展開し、+ ボタンを押します。
7. “libxml2.dylib” を選択し、[Add] ボタンを押します。
8. 使用したいアプリケーションの *.m ソースファイルで、Objective-C ヘッダーのインポート定義を記載します。下記を参照してください。

```
#import "ePOS-Device.h"
```

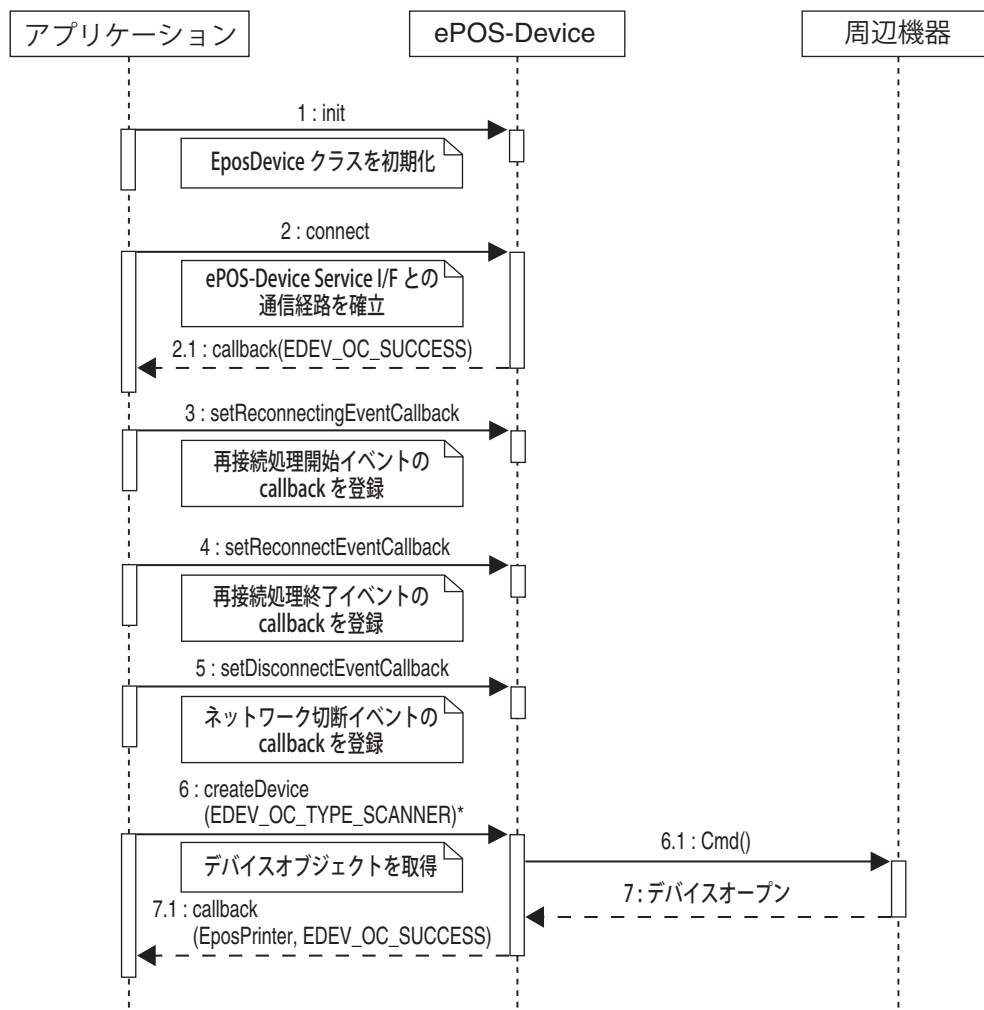
3

プログラミングフロー

ePOS-Device SDK for iOS の基本的なプログラミングのシーケンスは、以下のとおりです。

ネットワークを自動で再接続する場合

TM-i とのネットワーク通信が切断された場合、自動で再接続するシーケンスは、以下のとおりです。

参考	以下はバーコードスキャナー使用時を例にしています。

(1/2)

*: buffer パラメーターを、ネットワークの再接続時にデータを再送する設定にします。

(2/2)

アプリケーション
ePOS-Device
周辺機器
8 : デバイスに応じた callback の登録
9 : スキャン
ネットワークが切断
3.1 : 再接続処理開始イベント
再接続処理開始を通知
9.1: データ ()
ePOS-Device Service I/F で
データを保持
ネットワークの再接続が成功
4.1 : 再接続終了イベント
再接続終了を通知
9.1.1 : バーコードデータ検出イベント
ePOS-Device Service I/F で
保持したデータを送信
10: deleteDevice(EposScanner)
10.1 : Cmd()
デバイスオブジェクトを開放
11: デバイスクローズ
11.1 : callback(EDEV_OC_SUCCESS)
12: disconnect
ePOS-Device Service I/F との
通信経路を切断

コミュニケーションボックスのプログラミングフロー

コミュニケーションボックス使用した基本的なプログラミングのシーケンスは、以下のとおりです。
シーケンス図中の「クライアント」は、アプリケーションを指しています。

(1/3)

クライアント A
ePOS-Device
クライアント B
クライアント C
(2/3)
4 : openCommBox
コミュニケーションボックスをオープン
4.2 : クライアント B をメンバーに追加
4.3 : callback(EDEV_OC_SUCCESS)
5 : setReceiveEventCallback
4 : openCommBox
コミュニケーションボックスをオープン
4.2 : クライアント C をメンバーに追加
4.3 : callback(EDEV_OC_SUCCESS)
6 : sendData()
データを送信
6.1 : データを転送するクライアントを特定
5 : setReceiveEventCallback
6.2 : メッセージ受信イベント
データを転送
6.2 : メッセージ受信イベント
データを転送

(3/3)
クライアント A
ePOS-Device
クライアント B
クライアント C
7 : closeCommBox
7.1 : クライアント C をメンバーから削除
コミュニケーションボックスをクローズ
7.2 : callback(EDEV_OC_SUCCESS)
7 : closeCommBox()
コミュニケーションボックスをクローズ
7.1 : クライアント B をメンバーから削除
7.2 : callback(EDEV_OC_SUCCESS)
7 : closeCommBox
コミュニケーションボックスをクローズ
7.1 : クライアント A をメンバーから削除
7.2 : callback(EDEV_OC_SUCCESS)
8 : コミュニケーションボックスを破棄

ネットワークを自動で再接続する場合

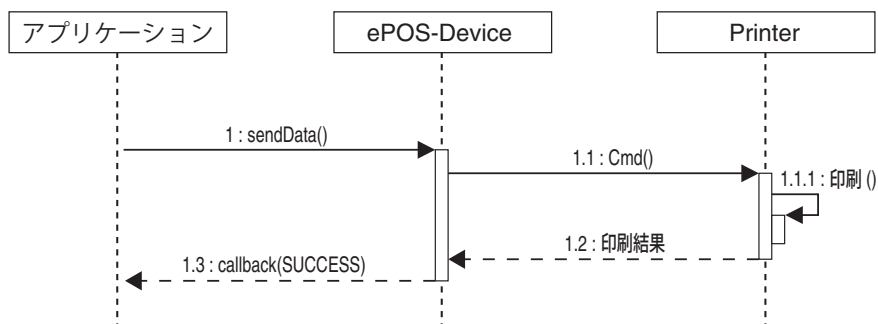
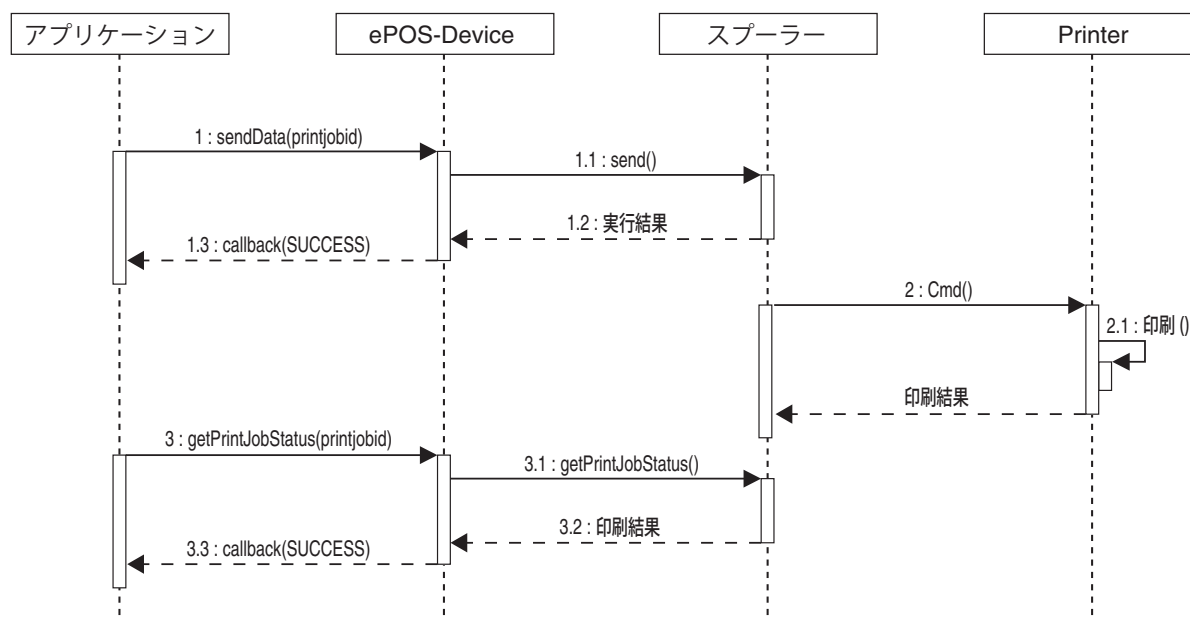
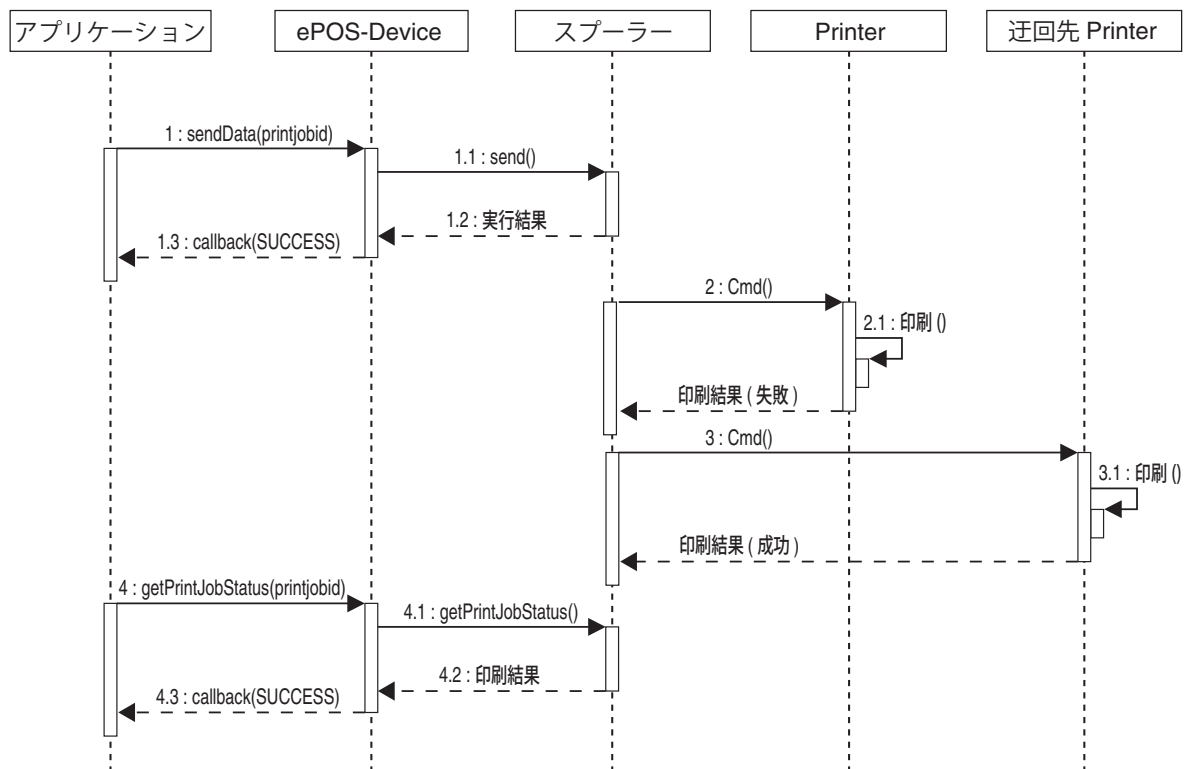
TM-i とのネットワーク通信が再接続処理が行われた場合、コミュニケーションボックスのシーケンスは、以下のとおりです。シーケンス図中の「クライアント」は、アプリケーションを指しています。

(1/2)

(2/2)

クライアント A
ePOS-Device
クライアント B
7 : openCommBox
コミュニケーションボックス
をオープン
callback(EDEV_OC_SUCCESS)
8 : setReceiveEventCallback
8 : setReceiveEventCallback
ネットワークが切断
9 : sendData()
コミュニケーションボックス
にメッセージを送信
3.2 : 再接続処理開始イベント
再接続処理開始を通知
ネットワークの再接続が成功
4.1 : 再接続処理終了イベント
再接続処理終了を通知
9.1 : メッセージ受信イベント
ネットワーク切断中に
受信したデータを転送
10 : closeCommBox
コミュニケーションボックス
をクローズ
callback(EDEV_OC_SUCCESS)
10 : closeCommBox
コミュニケーションボックス
をクローズ
callback(EDEV_OC_SUCCESS)

スプーラー機能を使用する場合

スプーラー機能を無効にした場合、アプリケーションから ePOS-Device Service I/F にリクエストを送信すると、印刷を直ちに実行し、印刷完了後にレスポンスをアプリケーションに返します。

スプーラー機能を有効にした場合、アプリケーションから ePOS-Device Service I/F にリクエストを送信すると、印刷データをキューに入れ、印刷完了を待たずにレスポンスをアプリケーションに返します。

この時、出力先のプリンターが印刷できない場合、ePOS-Device Service I/F は、アプリケーションにエラーを返しません。迂回印刷を有効にすると、代替プリンターで印刷でき、アプリケーションは、印刷結果も後で取得できます。以下のシーケンス図を参照してください。

API リファレンス

本章では、ePOS-Device SDK for iOS で用意されている API について説明しています。

ePOS-Device API 一覧

ePOS-Device SDK for iOS には、以下のクラスが用意されています。

- ❏ EposDevice クラス (33 ページ)
- ❏ EposDisplay クラス (34 ページ)
- ❏ EposKeyboard クラス (34 ページ)
- ❏ EposPrinter クラス (35 ページ)
- ❏ EposScanner クラス (37 ページ)
- ❏ EposSimpleSerial クラス (37 ページ)
- ❏ EposCommBoxManager クラス (37 ページ)
- ❏ EposCommBox クラス (37 ページ)
- ❏ EposDeviceLog クラス (37 ページ)

EposDevice クラス

API	説明	ページ
init	EposDevice オブジェクトの初期化	38
connect	通信経路の確立	39
disconnect	通信経路の切り離し	40
isConnected	通信経路の確立状態を取得	40
createDevice	デバイスオブジェクトの取得	41
deleteDevice	デバイスオブジェクトの破棄	43
getAdmin	管理者情報の取得	44
getLocation	設置場所情報の取得	44
getCommBoxManager	EposCommBoxManager オブジェクトを取得	44
setReconnectingEventCallback	再接続処理開始イベントのコールバックメソッドを登録	45
setReconnectEventCallback	再接続終了イベントのコールバックメソッドを登録	45
setDisconnectEventCallback	ネットワーク切断イベントのコールバックメソッドを登録	46

EposDisplay クラス

API	説明	ページ
sendData	制御命令の送信	47
clearCommandBuffer	命令バッファーをクリア	47
reset	リセット	48
createWindow	表示領域の定義	49
destroyWindow	表示領域の設定の破棄	50
setCurrentWindow	表示領域の切り替え	50
clearWindow	現在の表示領域を消去	51
setCursorPosition	カーソル位置の移動	51
moveCursorPosition	表示領域内でカーソル位置を移動	52
setCursorType	カーソル表示の変更	52
addText	テキストの表示	53
addReverseText	反転テキストの表示	54
addMarquee	マーキー表示	55
setBlink	表示の点滅	56
setBrightness	表示輝度の変更	56
showClock	時計の表示	57
addCommand	任意コマンドの実行	57
setReceiveEventCallback	制御結果受信イベントのコールバックメソッドを登録	58

EposKeyboard クラス

API	説明	ページ
setPrefix	連続した文字列として扱う条件を設定	59
setKeyPressEventCallback	キー押下検出イベントのコールバックメソッドを登録	60
setStringEventCallback	文字列検出イベントのコールバックメソッドを登録	61

EposPrinter クラス

API	説明	ページ
sendData	• 印刷ドキュメントの送信 • ジョブ ID を指定して印刷ドキュメントを送信 *	62
getPrintJobStatus *	ジョブ ID の印刷結果を取得	63
clearCommandBuffer	命令バッファーをクリア	63
addTextAlign	位置揃え設定を命令バッファーに追加	64
addTextLineSpace	改行量設定を命令バッファーに追加	65
addTextRotate	倒立印字設定を命令バッファーに追加	65
addText	文字印字を命令バッファーに追加	66
addTextLang	言語設定を命令バッファーに追加	67
addTextFont	文字フォント設定を命令バッファーに追加	68
addTextSmooth	文字スムージング設定を命令バッファーに追加	68
addTextDouble	文字倍角設定を命令バッファーに追加	69
addTextSize	文字倍率設定を命令バッファーに追加	70
addTextStyle	文字装飾設定を命令バッファーに追加	71
addTextPosition	文字印字位置設定を命令バッファーに追加	72
addTextVPosition	縦方向の印字開始位置設定を命令バッファーに追加	72
addFeedUnit	ドット単位の紙送りを命令バッファーに追加	73
addFeedLine	行単位の紙送りを命令バッファーに追加	73
addFeedPosition	ラベル紙・ブラックマーク紙の制御を命令バッファーに追加	74
addFeed	改行を命令バッファーに追加	74
addImage	ラスターイメージ印字を命令バッファーに追加	75
addLogo	NV ロゴ印字を命令バッファーに追加	77
addBarcode	バーコード印字を命令バッファーに追加	78
addSymbol	2 次元シンボル印字を命令バッファーに追加	82
addHLine	横罫線印字を命令バッファーに追加	87
addVLineBegin	縦罫線開始を命令バッファーに追加	88
addVLineEnd	縦罫線終了を命令バッファーに追加	89
addPageBegin	ページモード開始を命令バッファーに追加	90
addPageEnd	ページモード終了を命令バッファーに追加	90
addPageArea	ページモード印字領域設定を命令バッファーに追加	91
addPageDirection	ページモード印字方向設定を命令バッファーに追加	92
addPagePosition	ページモード印字位置設定を命令バッファーに追加	93
addPageLine	ページモード直線描画を命令バッファーに追加	94
addPageRectangle	ページモード四角形描画を命令	95
addCut	用紙カットを命令バッファーに追加	96
addPulse	ドロアーキックを命令バッファーに追加	97
addSound	ブザー鳴動を命令バッファーに追加	98
addLayout	用紙レイアウトの設定を命令バッファーに追加	100
addCommand	コマンドを命令バッファーに追加	103
addRecovery	エラーからの復帰タグを追加する	104
addReset	プリンターのリセットタグを追加する	104

4

API	説明	ページ
startMonitor	ステータスイベントの有効化	105
stopMonitor	ステータスイベントの無効化	105
recover	復帰可能エラーから復帰	106
reset	プリンターをリセット	106
halftone プロパティ	ラスターイメージのハーフトーン処理方法	107
brightness プロパティ	ラスターイメージの明るさ補正値	108
force プロパティ	強制送信モード	110
timeout プロパティ	送信タイムアウト時間	110
interval プロパティ	プリンターステータスの更新間隔	110
setReceiveEventCallback	• 応答ドキュメント受信イベントのコールバックメソッドを登録 • ジョブIDの応答ドキュメント受信イベントのコールバックメソッドを登録 *	111
setStatusChangeEventCallback	ステータス変更イベントのコールバックメソッドを登録	114
setBatteryStatusChangeEventCallback	バッテリーステータス変更イベントのコールバックメソッドを登録	115
setOnlineEventCallback	オンラインイベントのコールバックメソッドを登録	116
setOfflineEventCallback	オフラインイベントのコールバックメソッドを登録	117
setPowerOffEventCallback	無応答イベントのコールバックメソッドを登録	118
setCoverOkEventCallback	カバークローズイベントのコールバックメソッドを登録	119
setCoverOpenEventCallback	カバーオープンイベントのコールバックメソッドを登録	120
setPaperOkEventCallback	用紙ありイベントのコールバックメソッドを登録	121
setPaperNearEndEventCallback	用紙残量少イベントのコールバックメソッドを登録	122
setPaperEndEventCallback	用紙エンドイベントのコールバックメソッドを登録	123
setDrawerClosedEventCallback	ドロアークローズイベントのコールバックメソッドを登録	124
setDrawerOpenEventCallback	ドロアーオープンイベントのコールバックメソッドを登録	125
setBatteryLowEventCallback	バッテリー残量なしイベントのコールバックメソッドを登録	126
setBatteryOkEventCallback	バッテリー残量ありイベントのコールバックメソッドを登録	127

* TM-i ファームウェア Ver.4.1 以降対応

EposScanner クラス

API	説明	ページ
setDataEventCallback	バーコードデータ検出イベントのコールバックメソッドを登録	128

EposSimpleSerial クラス

API	説明	ページ
sendCommand	任意コマンドの送信	129
setCommandReplyEventCallback	任意コマンドの送信結果通知イベントのコールバックメソッドを登録	130

EposCommBoxManager クラス

API	説明	ページ
openCommBox	コミュニケーションボックスをオープン	131
closeCommBox	コミュニケーションボックスをクローズ	133

EposCommBox クラス

API	説明	ページ
getCommHistory	メッセージの送信履歴を取得	134
sendData	メッセージをコミュニケーションボックスに送信	136
setReceiveEventCallback	コミュニケーションボックスのメッセージの受信を通知イベントのコールバックメソッドを登録	138

EposDeviceLog クラス

API	説明	ページ
setLogSettings	ログ出力機能の設定	139

EposDevice クラス

init

EposDevice クラスのインスタンスを初期化します。

構文

```
- (id) init;
```

戻り値

初期化済みの EposDevice クラスインスタンスが返ります。

connect

TM-i の ePOS-Device Service I/F との接続経路を確立します。

構文

```
- (int) connect:(NSString *) ipAddress
                Callback:(SEL) callback
                Target:(NSObject *) target;
```

パラメーター

- ipAddress:　TM-i の IP アドレスを指定します。
- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_ILLEGAL	すでに通信が開始されているデバイスを再度通信開始しようとした。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *) ipAddress
                         Code:(int) code
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- code:　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	接続が成功した
EDEV_OC_ERR_TIMEOUT	タイムアウトが発生した

disconnect

connect によって接続された通信経路を切断します。

構文

```
- (int) disconnect;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_ILLEGAL	通信が開始されていない状態で、本 API が呼び出された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

isConnected

connect によって接続された通信経路の確立状態を取得します。

端末がスリープした場合、戻り値に正しい値を返さない場合があります。

構文

```
- (BOOL) isConnected;
```

戻り値

通信経路の確立状態を返します。

戻り値	説明
YES	接続済み
NO	未接続

createDevice

デバイスとのインターフェイスとなるデバイスオブジェクトを取得します。

注意

- 本 API の実行に成功すると、指定デバイスは排他的にロックされ、再度同じデバイスに createDevice を実行すると、"EDEV_OC_ERR_IN_USE" が返ります。デバイスがプリンターの場合、排他的にロックされないため、複数のデバイスオブジェクトを取得できます。
- 再接続処理失敗イベント発生の復帰処理で、すぐに createDevice を実行すると、"EDEV_OC_ERR_IN_USE" が発生します。この場合、"EDEV_OC_ERR_IN_USE" 以外が取得できるまで、再試行してください。

構文

```
- (int) createDevice:(NSString *)deviceId
                     DeviceType:(int)deviceType
                     Crypto:(int)crypto
                     Buffer:(int)buffer
                     Callback:(SEL)callback
                     Target:(NSObject *)target;
```

パラメーター

- deviceId:　デバイス ID を指定します。

参考

- 第 2 パラメーター(deviceType) に EDEV_OC_TYPE_DISPLAY を指定する場合、"local_display" を指定します。
- 第 2 パラメーター (deviceType) に EDEV_OC_TYPE_SIMPLE_SERIAL を指定する場合、"local_serial" を指定します。

- deviceType:　以下のいずれかのデバイス種類を指定します。

設定値	説明
EDEV_OC_TYPE_PRINTER	デバイスタイプをプリンターに指定
EDEV_OC_TYPE_DISPLAY	デバイスタイプをカスタマーディスプレイに指定
EDEV_OC_TYPE_KEYBOARD	デバイスタイプをキーボードに指定
EDEV_OC_TYPE_SCANNER	デバイスタイプをスキャナーに指定
EDEV_OC_TYPE_SIMPLE_SERIAL	デバイスタイプをシリアル通信デバイスに指定

- crypto:　通信データを暗号化の要否を指定します。

設定値	説明
EDEV_OC_FALSE	無効

- buffer:　再接続処理中にデバイスで発生したデータをバッファリングし、再接続時にデータの再送の要否を指定します。

設定値	説明
EDEV_OC_TRUE	有効
EDEV_OC_FALSE	無効

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_IN_USE	使用中のデバイスが存在する。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *) ipAddress
                       DeviceId:(NSString *)deviceId
                       DeviceType:(int)deviceType
                       DeviceObject:(id)deviceObject
                       Code:(int) code
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。
- deviceType: 本 API で指定した deviceType がセットされます。
- deviceObject: 本 API で指定した deviceType に対応したデバイスオブジェクトがセットされます。

deviceType	デバイスオブジェクト
EDEV_OC_TYPE_PRINTER	EposPrinter
EDEV_OC_TYPE_DISPLAY	EposDisplay
EDEV_OC_TYPE_KEYBOARD	EposKeyboard
EDEV_OC_TYPE_SCANNER	EposScanner
EDEV_OC_TYPE_SIMPLE_SERIAL	EposSimpleSerial

- code: 処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	デバイスオブジェクトの取得に成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_NOT_FOUND	デバイスが見つからない。
EDEV_OC_ERR_IN_USE	デバイスが使用中。
EDEV_OC_ERR_OPEN	デバイスのオープンに失敗した。
EDEV_OC_ERR_TYPE_INVALIED	デバイスタイプが違う。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

deleteDevice

デバイスを解放し、他のクライアントから使用可能な状態にします。

構文

```
- (int) deleteDevice:(id)deviceObject
                     Callback:(SEL)callback
                     Target:(NSObject *)
```

パラメーター

- deviceObject:　デバイスオブジェクトを指定します。
- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

4

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *) ipAddress
                        DeviceId:(NSString *)deviceId
                        Code:(int) code
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- code:　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	デバイスのクローズに成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_NOT_OPENED	デバイスがオープンされていない。
EDEV_OC_ERR_CLOSE	デバイスのクローズに失敗した。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

getAdmin

TM-i に設定された管理者名を取得します。

> **参考**
> 管理者名は EPSON TMNet WebConfig によって設定できます。

構文

```
- (NSString *)getAdmin;
```

戻り値

管理者名の文字列が返されます。

getLocation

TM-i に設定された設置場所情報を取得します。

> **参考**
> 設置場所情報は EPSON TMNet WebConfig によって設定できます。

構文

```
- (NSString *)getLocation;
```

戻り値

設置場所情報が返されます。

getCommBoxManager

アプリケーション間通信を行う EposCommBoxManager オブジェクトを取得します。

構文

```
- (EposCommBoxManager *)getCommBoxManager;
```

setReconnectingEventCallback

再接続処理が開始されたことを受け取る、コールバックメソッドを登録します。

構文

```
-(void) setReconnectingEventCallback:(SEL)callback
                                   Target:(NSObject*)target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *) ipAddress
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。

setReconnectEventCallback

再接続処理が終了したことを受け取る、コールバックメソッドを登録します。

構文

```
-(void) setReconnectEventCallback:(SEL)callback
                                   Target:(NSObject*)target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *) ipAddress
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。

setDisconnectEventCallback

再接続処理が失敗したことを受け取る、コールバックメソッドを登録します。
コールバックメソッドには、再度接続を確立するなど、ネットワーク切断対策のための処理を行ってください。

構文

```
-(void) setDisconnectEventCallback:(SEL)callback
                        Target:(NSObject*)target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

コールバックメソッドの定義

- （void） *メソッドの名称*：（NSString *） ipAddress

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。

EposDisplay クラス

sendData

命令バッファー内の命令を送信します。

構文

```
- (int) sendData;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

clearCommandBuffer

命令バッファーをクリアします。

構文

```
- (int) clearCommandBuffer;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。

reset

カスタマーディスプレイの初期化設定を命令バッファーに追加します。初期化により、以下の状態になります。

- 表示文字がすべて消去され、登録されたすべてのウィンドウが破棄されます。
- カーソルはデフォルト状態に戻り、カスタマーディスプレイ原点に移動します。
- 表示点滅、表示輝度がデフォルト状態に戻ります。

構文

```
- (int)reset;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

createWindow

ウィンドウ設定を命令バッファーに追加します。

> **注意**
> ウィンドウは定義済みウィンドウと重ならないように定義してください。
> また、カスタマーディスプレイのサイズ（20 桁 ×2 行）に収まる大きさを指定してください。

構文

```
- (int)createWindow:(long)number X:(long)x Y:(long)y
                    Width:(long)width Height:(long)height
                    ScrallMode:(int)scrallMode;
```

パラメーター

- number:　ウィンドウ番号を指定します。1 ～ 4 から指定してください。
- x:　ウィンドウ原点のカスタマーディスプレイ上での x 座標を指定します。
  1 ～ 20 から指定してください。
- y:　ウィンドウ原点のカスタマーディスプレイ上での y 座標を指定します。
  1 ～ 2 から指定してください。
- width:　ウィンドウの幅を指定します。1 ～ 20 から指定してください。
- height:　ウィンドウの高さを指定します。1 ～ 2 から指定してください。
- scrollMode:　ウィンドウのスクロールモードの指定をします。以下から選択します。

設定値	説明
EDEV_OC_SCROLL_OVERWRITE	表示位置が上段右端位置にあるとき文字表示すると、表示位置を下段左端に移動し、下段右端にあるとき文字表示すると表示位置を上段左端に移動する
EDEV_OC_SCROLL_VERTICAL	表示位置が上段右端位置にあるとき文字表示すると、表示位置を下段左端に移動し、下段右端にあるとき文字表示するとすでに表示してある下段の表示文字を上段にスクロールし、下段の表示をクリアする
EDEV_OC_SCROLL_HORIZONTAL	表示位置が右端位置にあるとき文字表示すると、カーソルのある行にすでに表示しているすべての表示文字は 1 文字分左へスクロールし、新たな表示文字を右端に表示する

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

destroyWindow

定義済みウィンドウの破棄設定を命令バッファーに追加します。

構文

```
- (int)destroyWindow:(long)number;
```

パラメーター

- number:　破棄するウィンドウ番号を指定します。1 ～ 4 から指定してください。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setCurrentWindow

カレントウィンドウの変更設定を命令バッファーに追加します。
指定ウィンドウをカレントウィンドウとし、カーソルをカレントウィンドウの原点位置に移動させます。

構文

```
- (int)setCurrentWindow:(long)number;
```

パラメーター

- number:　移動先のウィンドウ番号を指定します。1 ～ 4 から指定してください。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

clearWindow

カレントウィンドウ内の表示消去を命令バッファーに追加します。

構文

```
- (int)clearWindow;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setCursorPosition

カーソル位置設定を命令バッファーに追加します。

構文

```
- (int)setCursorPosition:(long)x Y:(long)y;
```

パラメーター

- x:　　移動先となる x 座標を指定します。1 ～ 20 から指定します。
- y:　　移動先となる y 座標を指定します。1 か 2 を指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

moveCursorPosition

カレントウィンドウ内でのカーソル位置設定を命令バッファーに追加します。

構文

```
- (int)moveCurrentPosition:(int)position;
```

パラメーター

- position:　カレントウィンドウ内でのカーソルの移動先を指定します。

設定値	説明
EDEV_OC_MOVE_TOP_LEFT	移動先を左上端に設定
EDEV_OC_MOVE_TOP_RIGHT	移動先を右上端に設定
EDEV_OC_MOVE_BOTTOM_LEFT	移動先を左下端に設定
EDEV_OC_MOVE_BOTTOM_RIGHT	移動先を右下端に設定

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setCursorType

カーソル表示設定を命令バッファーに追加します。

構文

```
- (int) setCursorType:(int) type;
```

パラメーター

- type:　カーソルの表示方法を指定します。

設定値	説明
EDEV_OC_CURSOR_NONE	カーソル表示なし
EDEV_OC_CURSOR_UNDERLINE	アンダーライン

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addText

文字列の表示を命令バッファーに追加します。

構文

```
❏ - (int) addText:(NSString *)data;

❏ - (int) addText:(NSString *)data Lang:(int)lang;

❏ - (int) addText:(NSString *)data X:(long)x Y:(long)y;

❏ - (int) addText:(NSString *)data X:(long)x Y:(long)y
                     Lang:(int)lang;
```

パラメーター

- data:　表示文字列を指定します。
- lang:　文字列の表示言語を指定します。

設定値	説明
EDEV_OC_LANG_EN(デフォルト)	表示言語を英語に設定
EDEV_OC_LANG_JA	表示言語を仮名に設定

- x:　表示位置の x 座標を指定します。1 ～ 20 から指定します。
- y:　表示位置の y 座標を指定します。1 か 2 を指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addReverseText

反転文字列の表示を命令バッファーに追加します。

構文

```
❏ - (int) addReverseText:(NSString *)data;

❏ - (int) addReverseText:(NSString *)data Lang:(int)lang;

❏ - (int) addReverseText:(NSString *)data X:(long)x
                               Y:(long)y;

❏ - (int) addReverseText:(NSString *)data X:(long)x
                               Y:(long)y Lang:(int)lang;
```

パラメーター

- data: 表示文字列を指定します。
- lang: 文字列の表示言語を指定します。

設定値	説明
EDEV_OC_LANG_EN(デフォルト)	表示言語を英語に設定
EDEV_OC_LANG_JA	表示言語を仮名に設定

- x: 表示位置の x 座標を指定します。1 ～ 20 から指定します。
- y: 表示位置の y 座標を指定します。1 か 2 を指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addMarquee

マーキー表示を命令バッファーに追加します。

マーキー表示は、表示ウィンドウのスクロールモード設定とは無関係に、1 行の水平スクロールモードで表示されます。

構文

```
- (int)addMarquee:(NSString *)data Format:(int)format
                  UnitWait:(long)unitWait
                  RepeatWait:(long)repeatWait
                  RepeatCount:(long)repeatCount
                  Lang:(int)lang;
```

パラメーター

- data:　表示文字列を指定します。
- format:　文字列の表示方法を指定します。

設定値	説明
EDEV_OC_MARQUEE_WALK	ウィンドウ右端から文字列を表示する
EDEV_OC_MARQUEE_PLACE	ウィンドウ左端から文字列を表示する

- unitWait:　1 文字ごとの表示間隔 (ms) を指定します。指定可能な値は 0 ～ 2,000 です。
- repeatWait:　文字列表示の繰り返し間隔 (ms) を指定します。指定可能な値は 100 ～ 2,000 です。
- repeatCount:　文字列表示の繰り返し回数を指定します。指定可能な値は 0 ～ 127 です。
  0 を指定すると無制限に繰り返します。
- lang:　文字列の表示言語を指定します。

設定値	説明
EDEV_OC_LANG_EN(デフォルト)	表示言語を英語に設定
EDEV_OC_LANG_JA	表示言語を仮名に設定

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setBlink

画面の点滅設定を命令バッファーに追加します。

構文

```
- (int)setBlink:(long)interval;
```

パラメーター

- interval:　点滅時間 (ms) を指定します。0 ～ 12,700 から指定します。
  設定値は 50ms 単位に丸められます。0 を指定すると点灯状態となります。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setBrightness

カスタマーディスプレイの輝度設定を命令バッファーに追加します。

構文

```
- (int)setBrightness:(int)brightness;
```

パラメーター

- brightness:　カスタマーディスプレイの輝度値を指定します。

設定値	説明
EDEV_OC_BRIGHTNESS_20	カスタマーディスプレイの輝度値を 20%に設定する
EDEV_OC_BRIGHTNESS_40	カスタマーディスプレイの輝度値を 40%に設定する
EDEV_OC_BRIGHTNESS_60	カスタマーディスプレイの輝度値を 60%に設定する
EDEV_OC_BRIGHTNESS_100	カスタマーディスプレイの輝度値を 100%に設定する

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

showClock

時刻の表示を命令バッファーに追加します。

時刻表示を行うと、表示中の文字列はすべて消去されます。また時刻表示中に、他の命令を送信すると、時刻表示を終了します。

構文

```
- (int)showClock;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addCommand

コマンドを命令バッファーに追加します。

ESC/POS コマンドの詳細については ESC/POS アプリケーションプログラミングガイドを参照してください。ESC/POS アプリケーションプログラミングガイドの入手には弊社との機密保持契約が必要です。詳しくは販売元 までお問い合わせください。

構文

```
- (int) addCommand:(NSData *)data;
```

パラメーター

- data：　ESC/POS コマンドをバイナリーデータで指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setReceiveEventCallback

応答ドキュメント受信イベントを受け取る、コールバックメソッドを登録します。

> **参考**　本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setReceiveEventCallback: (SEL) callback
                         Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**　本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                DeviceId:(NSString *)deviceId
                Success:(int)success
                Code:(int)code
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- success:　表示の実行結果がセットされます。

エラーステータス	説明
EDEV_OC_TRUE	表示が成功した。
EDEV_OC_FALSE	表示が失敗した。

- code:　処理の実行結果がセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_NOT_FOUND	デバイスが見つからない。
EDEV_OC_ERR_OPEN	デバイスのオープンに失敗した。
EDEV_OC_ERR_INVALID_WINDOW	未登録のウィンドウが指定された。
EDEV_OC_ERR_PORT	デバイスとの内部通信エラーが発生した。
EDEV_OC_ERR_TIMEOUT	デバイスとの通信でタイムアウトエラーが発生した。
EDEV_OC_ERR_PARAM	不正なパラメーターを検出した。

EposKeyboard クラス

setPrefix

キーボード入力を文字列としてまとめて受け取るさい、文字列の始まりと判断するためのキーコードを設定します。

参考

nil、または ""(空文字) を入力した場合、設定が解除されます。

構文

```
- (int)setPrefix:(NSData *)data;
```

パラメーター

- data:　キーコードの配列を指定します。キーコードは 147 ページ「キーコード一覧」を参照してください。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setKeyPressEventCallback

キー押下イベントを受け取る、コールバックメソッドを登録します。

参考	本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setKeyPressEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考	本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                  DeviceId:(NSString *)deviceId
                  KeyCode:(int)keyCode
                  Ascii:(NSString *)ascii;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- keyCode:　押下されたキーコードがセットされます。
- ascii:　キーコードに対応する文字がセットされます。
  キーコードに対応する文字がない場合、” ”（空文字）がセットされます。

setStringEventCallback

setPrefix (59 ページ) で指定した文字群のいずれかで始まり、Enter キーを検出するまでの間に検出した文字入力を文字列として受け取る、コールバックメソッドを登録します。

> **参考**　本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setStringEventCallback: (SEL) callback
                       Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**　本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                 DeviceId:(NSString *)deviceId
                 Input:(NSString *)input
                 Prefix:(int)prefix;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- input:　検出された文字列がセットされます。
- prefix:　文字列の開始と判断したキーコードがセットされます。

sendData

命令バッファーを送信します。ジョブ ID を指定して送信することもできます。

参考	• 本 API の実行結果は、setReceiveEventCallback (111 ページ) に返されます。 • 本 API を実行すると、force プロパティ (109 ページ) の設定が、EDEV_OC_FALSE になります。

構文

❏ - (int) ***sendData***;

❏ ジョブ ID 指定用

```
- (int) sendData:(NSString*)printJobId;
```

注意	ジョブ ID 指定用の構文は、TM-i ファームウェア Ver.4.1 以降で対応しています。 詳細は、アプリケーションから印刷ジョブ ID を指定 (12 ページ) を参照してください。

パラメーター

• printJobId: ジョブ ID を指定します。
使用可能な文字は 1 ～ 30 文字の英数字、アンダースコア、ハイフン、ピリオドです。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

getPrintJobStatus

指定したジョブ ID の印刷結果を取得します。(TM-i ファームウェア Ver.4.1 以降)

- 本 API の実行結果は、setReceiveEventCallback (111 ページ)に返されます。
- 同じジョブ ID で、複数の印刷処理を実行した場合、一番新しい印刷ジョブのステータスを取得します。

構文

```
- (int) getPrintJobStatus:(NSString*)printJobId;
```

パラメーター

- printJobId：　取得したいジョブ ID を指定します。
  使用可能な文字は 1 〜 30 文字の英数字、アンダースコア、ハイフン、ピリオドです。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

clearCommandBuffer

命令バッファーをクリアします。

構文

```
- (int) clearCommandBuffer;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。

addTextAlign

位置揃え設定を命令バッファーに追加します。

- 本 API の設定は、NV ロゴ / バーコード /2 次元シンボルにも適用されます。
- ページモードでは、本 API の指定は無効です。ページモードで横方向印字位置を指定するには、addTextPosition (72 ページ) を使います。

構文

```
- (int) addTextAlign:(int)align;
```

パラメーター

- align :　位置揃えを指定します。

設定値	説明
EDEV_OC_ALIGN_LEFT (デフォルト)	左揃え
EDEV_OC_ALIGN_CENTER	中央揃え
EDEV_OC_ALIGN_RIGHT	右揃え

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextLineSpace

改行量設定を命令バッファーに追加します。

構文

```
- (int) addTextLineSpace:(long)linespc;
```

パラメーター

- linespc：　改行量 (ドット単位) を指定します。0 ～ 255 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextRotate

倒立印字設定を命令バッファーに追加します。

- 本 API の設定は、バーコード /2 次元シンボルにも適用されます。
- ページモードでは、本 API の指定は無効です。ページモードの場合、addPageDirection (92 ページ) を使います。

構文

```
- (int) addTextRotate:(int)rotate;
```

パラメーター

- rotate：　倒立印字の有無を指定します。

設定値	説明
EDEV_OC_TRUE	倒立印字を指定
EDEV_OC_FALSE (デフォルト)	倒立印字を解除

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addText

文字の印字を命令バッファーに追加します。

注意	テキストの印字後、テキスト以外を印字する場合、改行または紙送りを実行してください。
参考	ページモードでは、文字のベースラインドット（148 ページ「プリンターの仕様」）を基準として、現在の印字位置に文字を展開します。

構文

```
- (int) addText:(NSString *)data;
```

パラメーター

- data：　印字する文字列を指定します。
  水平タブ / 改行は、以下のエスケープシーケンスを使用します。

文字列	説明
\t	水平タブ (HT)
\n	改行 (LF)
\\	バックスラッシュ

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextLang

言語設定を命令バッファーに追加します。
本 API で指定された言語設定に従って、addText (66 ページ) で指定された文字列をエンコードします。

構文

```
- (int) addTextLang:(int)lang;
```

パラメーター

- lang：　対象言語を指定します。

設定値	言語
EDEV_OC_LANG_EN (デフォルト)	英語 (ANK 仕様)
EDEV_OC_LANG_JA	日本語 (国際文字セットも日本に変わります。)

参考

印字できる文字コードは、各プリンターの詳細取扱説明書を参照してください。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextFont

文字のフォント設定を命令バッファーに追加します。

構文

```
- (int) addTextFont:(int)font;
```

パラメーター

- font： フォントを指定します。

設定値	言語
EDEV_OC_FONT_A（デフォルト）	フォント A
EDEV_OC_FONT_B	フォント B
EDEV_OC_FONT_C	フォント C
EDEV_OC_FONT_D	フォント D
EDEV_OC_FONT_E	フォント E

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextSmooth

スムージング設定を命令バッファーに追加します。

構文

```
- (int) addTextSmooth:(int)smooth;
```

パラメーター

- smooth： スムージングの有無を指定します。

設定値	説明
EDEV_OC_TRUE	スムージングを指定
EDEV_OC_FALSE（デフォルト）	スムージングを解除

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextDouble

文字の倍角設定を命令バッファーに追加します。

構文

```
- (int) addTextDouble:(int)dw Dh:(int)dh;
```

パラメーター

- dw：　文字の横倍角を指定します。

設定値	説明
EDEV_OC_TRUE	横倍角を指定
EDEV_OC_FALSE (デフォルト)	横倍角を解除
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

- dh：　文字の縦倍角を指定します。

設定値	説明
EDEV_OC_TRUE	縦倍角を指定
EDEV_OC_FALSE (デフォルト)	縦倍角を解除
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

参考

- dw と dh のパラメーターの両方を EDEV_OC_TRUE にした場合、4 倍角の文字が印字されます。
- すべてのパラメーターに EDEV_OC_PARAM_UNSPECIFIED を指定した場合、戻り値に EDEV_OC_ERR_PARAM が返ります。

4

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextSize

文字の倍率設定を命令バッファーに追加します。

構文

```
- (int) addTextSize:(long)width Height:(long)height;
```

パラメーター

- width : 文字の横倍率を指定します。

設定値	説明
1 ～ 8 の整数	横方向の倍率を指定 (デフォルト : 1)
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

- height : 文字の縦倍率を指定します。

設定値	説明
1 ～ 8 の整数	縦方向の倍率を指定 (デフォルト : 1)
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

参考 すべてのパラメーターに EDEV_OC_PARAM_UNSPECIFIED を指定した場合、戻り値に EDEV_OC_ERR_PARAM が返ります。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextStyle

文字の装飾設定を命令バッファーに追加します。

構文

```
- (int) addTextStyle:(int)reverse Ul:(int)ul
                      Em:(int)em Color:(int)color;
```

パラメーター

- reverse：　白黒反転文字を指定します。

設定値	説明
EDEV_OC_TRUE	白黒反転文字を指定
EDEV_OC_FALSE (デフォルト)	白黒反転文字を解除
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

- ul：　アンダーラインを指定します。

設定値	説明
EDEV_OC_TRUE	アンダーラインを指定
EDEV_OC_FALSE (デフォルト)	アンダーラインを解除
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

- em：　太字を指定します。

設定値	説明
EDEV_OC_TRUE	太字を指定
EDEV_OC_FALSE(デフォルト)	太字を解除
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

- color：　色を指定します。

設定値	説明
EDEV_OC_COLOR_NONE	非印字
EDEV_OC_COLOR_1 (デフォルト)	第 1 色
EDEV_OC_COLOR_2	第 2 色
EDEV_OC_COLOR_3	第 3 色
EDEV_OC_COLOR_4	第 4 色
EDEV_OC_PARAM_UNSPECIFIED	設定を変更しない

参考

すべてのパラメーターに EDEV_OC_PARAM_UNSPECIFIED を指定した場合、戻り値に EDEV_OC_ERR_PARAM が返ります。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextPosition

横方向の印字開始位置を命令バッファーに追加します。

構文

```
- (int) addTextPosition:(long)x;
```

パラメーター

- x：　横方向の印字開始位置（ドット単位）を指定します。
0 ～ 65535 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addTextVPosition

縦方向の印字開始位置を命令バッファーに追加します。

参考

本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (long) addTextVPosition:(long)y;
```

パラメーター

- y：　縦方向の印字開始位置（ドット単位）を指定します。
0 ～ 65535 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addFeedUnit

ドット単位の紙送りを命令バッファーに追加します。

構文

```
- (int) addFeedUnit:(long)unit;
```

パラメーター

- unit：　紙送り量（ドット単位）を指定します。0 ～ 255 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addFeedLine

行単位の紙送りを命令バッファーに追加します。

構文

```
- (int) addFeedLine:(long)line;
```

パラメーター

- line：　紙送り量（行単位）を指定します。0 ～ 255 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addFeedPosition

ラベル紙 / ブラックマーク紙の紙送り位置の指定を命令バッファーに追加します。

参考
- ラベル紙 / ブラックマーク紙の制御はスタンダードモードで行ってください。
- ページモードでは、本 API の設定は無効です。

構文

```
- (int) addFeedPosition:(int)position;
```

パラメーター

- position： ラベル紙 / ブラックマーク紙の紙送り位置を指定します。

設定値	説明
EDEV_OC_FEED_PEELING	剥離位置までの紙送り
EDEV_OC_FEED_CUTTING	カット位置までの紙送り
EDEV_OC_FEED_CURRENT_TOF	現在ラベルの頭出し位置までの紙送り
EDEV_OC_FEED_NEXT_TOF	次ラベルの頭出し位置までの紙送り

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addFeed

改行を命令バッファーに追加します。

構文

```
- (int) addFeed;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addImage

ラスターイメージの印字を命令バッファーに追加します。

UIImage クラスのグラフィックを印字します。

UIImage クラスのグラフィックのうち、指定範囲を本 API のカラーモードの設定、halftone プロパティ（107 ページ）、brightness プロパティ（108 ページ）の設定に従って、ラスターイメージデータに変換します。画像の 1 ピクセルがプリンターの 1 ドットに相当します。透明色が含まれている場合、画像の背景を白とみなします。

参考

- ラスターイメージを高速に印字する場合、addTextAlign メソッド（64 ページ）を ALIGN_LEFT に指定し、本 API の width パラメーターの値をプリンターの用紙幅を超えない 8 の倍数に指定してください。
- ページモードでは、ラスターイメージの左下ドットを基準として、現在の印字位置にラスターイメージを展開します。印字位置は移動しません。

注意

- ページモードでは多階調印字をサポートしていません。スタンダードモードでのみ多階調グラフィックスの印字が可能です。

構文

```
- (int) addImage:(UIImage *)data X:(long)x
                  Y:(long)y Width:(long)width
                  Height:(long)height Color:(int)color
                  Mode:(int)mode;
```

パラメーター

- data：　UIImage クラスのインスタンスを指定します。
- x：　印字範囲の横方向の開始位置を指定します。0 ～ 65535 の整数値で指定します。
- y：　印字範囲の縦方向の開始位置を指定します。0 ～ 65535 の整数値で指定します。
- width：　印字範囲の幅を指定します。1 ～ 65535 の整数値で指定します。
- height：　印字範囲の高さを指定します。1 ～ 65535 の整数値で指定します。
- color：　色を指定します。

設定値	説明
EDEV_OC_COLOR_NONE	非印字
EDEV_OC_COLOR_1 (デフォルト)	第 1 色
EDEV_OC_COLOR_2	第 2 色
EDEV_OC_COLOR_3	第 3 色
EDEV_OC_COLOR_4	第 4 色
EDEV_OC_PARAM_DEFAULT	既定値 (第 1 色)

- mode：　カラーモードを指定します。

設定値	説明
EDEV_OC_MODE_MONO	モノクロ (2 階調)
EDEV_OC_MODE_GRAY16	多階調 (16 階調)
EDEV_OC_PARAM_DEFAULT	モノクロ (2 階調)

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

参考

x/y と width/height で指定された領域が、data で指定した画像のサイズに収まっていない場合、戻り値に EDEV_OC_ERR_PARAM が返される。

addLogo

NV ロゴの印字を命令バッファーに追加します。
プリンターの NV メモリーに登録されているロゴを印字します。

参考	• ロゴは以下のユーティリティーを使って、あらかじめプリンターにロゴの登録します。 ＊ 機種専用ユーティリティー ＊ ロゴ登録ユーティリティー（TMFLogo） • ページモードでは、ロゴの左下ドットを基準として、現在の印字位置にロゴを展開します。
注意	ページモードでは多階調印字をサポートしていません。スタンダードモードでのみ多階調グラフィックスの印字が可能です。

構文

```
- (int) addLogo:(long)key1 Key2:(long)key2;
```

パラメーター

- key1：　NV ロゴのキーコード 1 を指定します。0 ～ 255 の整数値で指定します。
- key2：　NV ロゴのキーコード 2 を指定します。0 ～ 255 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addBarcode

バーコード印字を命令バッファーに追加します。

> **参考** ページモードでは、バーコードの左下ドット（HRIを除く）を基準として、現在の印字位置にバーコードを展開します。

構文

```
- (int) addBarcode:(NSString *)data Type:(int)type
                    Hri:(int)hri Font:(int)font
                    Width:(long)width Height:(long)height;
```

パラメーター

- data：　バーコードデータを文字列で指定します。

> **参考** typeで指定するバーコードの規格にしたがった文字列を指定してください。
> 規格にしたがっていない場合、バーコードは印刷されません。

種類	説明
UPC-A	11桁の数字を指定した場合、チェックデジットを自動で付加します。 12桁の数字を指定した場合、12桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
UPC-E	最初の桁に0を指定してください。 2～6桁目にメーカーコードを指定してください。 7～11桁目にアイテムコードを右詰めで指定してください。アイテムコードの桁数はメーカーコードにより異なります。使用しない桁は0を指定してください。 11桁の数字を指定した場合、チェックデジットを自動で付加します。 12桁の数字を指定した場合、12桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
EAN13 JAN13	12桁の数字を指定した場合、チェックデジットを自動で付加します。 13桁の数字を指定した場合、13桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
EAN8 JAN8	7桁の数字を指定した場合、チェックデジットを自動で付加します。 8桁の数字を指定した場合、8桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
CODE39	先頭の文字が"*"の場合、この文字をスタートキャラクターとして処理します。"*"以外の場合、スタートキャラクターを自動で付加します。
ITF	スタートコードおよびストップコードを自動で付加します。 チェックデジットの付加および検算は行いません。

<table>
<tr><th>種類</th><th>説明</th></tr>
<tr><td>CODABAR</td><td>スタートキャラクター (A ～ D, a ～ d) を
指定してください。
ストップキャラクター (A ～ D, a ～ d) を
指定してください。
チェックデジットの付加および検算は行いません。</td></tr>
<tr><td>CODE93</td><td>スタートキャラクターおよびストップキャラクターを自動で付加します。
チェックデジットを計算して自動で付加します。</td></tr>
<tr><td>CODE128</td><td>スタートキャラクター (CODE A, CODE B, CODE C) を指定してください。
ストップキャラクターを自動で付加します。
チェックデジットを計算して自動で付加します。
以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。
FNC1: {1
FNC2: {2
FNC3: {3
FNC4: {4
CODE A: {A
CODE B: {B
CODE C: {C
SHIFT: {S
{: {{</td></tr>
<tr><td>GS1-128</td><td>スタートキャラクター、FNC1、チェックデジット、ストップキャラクターを自動で付加します。
アプリケーション識別子 (AI) とそれに続くデータのチェックデジットを計算して自動で付加するには、チェックデジットの位置に文字 "*" を指定します。
アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。
アプリケーション識別子 (AI) とデータの間に空白を挿入することができます。空白は HRI の印字文字として使用し、データとしてエンコードしません。
以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。
FNC1: {1
FNC3: {3
(: {(<br>): {)
: {
{: {{</td></tr>
<tr><td>GS1 DataBar Omnidirectional</td><td rowspan="3">アプリケーション識別子 (AI) とチェックデジットを除く 13 桁の商品識別番号 (GTIN) を指定してください。</td></tr>
<tr><td>GS1 DataBar Truncated</td></tr>
<tr><td>GS1 DataBar Limited</td></tr>
</table>

種類	説明
GS1 Databar Expanded	アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。 以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。 FNC1: {1 (: {( ): {)

文字列で表現できないバイナリーデータを指定する場合、
以下のエスケープシーケンスで指定します。

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

- type : バーコードの種類を指定します。

設定値	種類
EDEV_OC_BARCODE_UPC_A	UPC-A
EDEV_OC_BARCODE_UPC_E	UPC-E
EDEV_OC_BARCODE_EAN13	EAN13
EDEV_OC_BARCODE_JAN13	JAN13
EDEV_OC_BARCODE_EAN8	EAN8
EDEV_OC_BARCODE_JAN8	JAN8
EDEV_OC_BARCODE_CODE39	CODE39
EDEV_OC_BARCODE_ITF	ITF
EDEV_OC_BARCODE_CODABAR	CODABAR
EDEV_OC_BARCODE_CODE93	CODE93
EDEV_OC_BARCODE_CODE128	CODE128
EDEV_OC_BARCODE_GS1_128	GS1-128
EDEV_OC_BARCODE_GS1_DATABAR_OMNIDIRECTIONAL	GS1 DataBar Omnidirectional
EDEV_OC_BARCODE_GS1_DATABAR_TRUNCATED	GS1 DataBar Truncated
EDEV_OC_BARCODE_GS1_DATABAR_LIMITED	GS1 DataBar Limited
EDEV_OC_BARCODE_GS1_DATABAR_EXPANDED	GS1 Databar Expanded

- hri : HRI の位置を指定します。

設定値	説明
EDEV_OC_HRI_NONE(デフォルト)	印字しない
EDEV_OC_HRI_ABOVE	バーコードの上
EDEV_OC_HRI_BELOW	バーコードの下
EDEV_OC_HRI_BOTH	バーコードの上と下の両方

- font：　HRI フォントを指定します。

設定値	言語
EDEV_OC_FONT_A(デフォルト)	フォント A
EDEV_OC_FONT_B	フォント B
EDEV_OC_FONT_C	フォント C
EDEV_OC_FONT_D	フォント D
EDEV_OC_FONT_E	フォント E

- width：　1 モジュールの幅をドット単位で指定します。2 ～ 6 の整数値で指定します。
- height：　バーコードの高さをドット単位で指定します。1 ～ 255 の整数値で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addSymbol

２次元シンボル印字を命令バッファーに追加します。

> **参考** ページモードでは、2次元シンボルの左上ドットを基準として、現在の印字位置にバーコードを展開します。

構文

```
- (int) addSymbol:(NSString *)data Type:(int)type
                  Level:(int)level Width:(long)width
                  Height:(long)height Size:(long)size;
```

パラメーター

- data：　２次元シンボルデータを文字列で指定します。

> **参考** type で指定する 2 次元シンボルの規格にしたがった文字列を指定してください。
> 規格にしたがっていない場合、2 次元シンボルは印刷されません。

文字列	説明
Standard PDF417	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 データ領域の最大コードワード数は 928 個、1 段あたりのデータ領域の最大コードワード数は 30 個、最大段数は 90 段です。
Truncated PDF417	
QR Code Model 1	文字列をシフト JIS に変換後、エスケープシーケンスの処理を行い、データの種類を以下の中から選択してエンコードします。 数字：　0 ～ 9 英数字：　0 ～ 9, A ～ Z, スペース , $, %, *, +, -, ., /, : 漢字：　シフト JIS 値 8 ビットバイトデータ： 0x00 ～ 0xff
QR Code Model 2	
Micro QR Code	

文字列	説明
MaxiCode Mode 2	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 モード 2 およびモード 3 の場合、最初のデータが ()>\x1e01\x1dyy (yy は 2 桁の数字) の場合、これをメッセージヘッダーとして処理し、次のデータからプライマリメッセージとして処理します。それ以外の場合、最初のデータからプライマリメッセージとして処理します。 モード 2 の場合、プライマリメッセージを以下の形式で指定してください。 郵便コード (1 ～ 9 桁の数字) GS:(\x1d) ISO 国名コード (1 ～ 3 桁の数字) GS:(\x1d) サービスクラスコード (1 ～ 3 桁の数字) モード 3 の場合、プライマリメッセージを以下の形式で指定してください。 郵便コード (1 ～ 6 個のコードセット A で変換可能なデータ) GS(\x1d) ISO 国名コード (1 ～ 3 桁の数字) GS(\x1d) サービスクラスコード (1 ～ 3 桁の数字)
MaxiCode Mode 3	
MaxiCode Mode 4	
MaxiCode Mode 5	
MaxiCode Mode 6	
GS1 DataBar Stacked	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 アプリケーション識別子 (AI) とチェックデジットを除く 13 桁の商品識別番号 (GTIN) を指定してください。
GS1 DataBar Stacked Omnidirectional	
GS1 DataBar Expanded Stacked	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。 以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。 FNC1:　{1 (:　{( ):　{)
Aztec Code	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。
DataMatrix	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。

文字列で表現できないバイナリーデータを指定する場合、以下のエスケープシーケンスで指定します。

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

- type： 2次元シンボルの種類を指定します。

設定値	種類
EDEV_OC_SYMBOL_PDF417_STANDARD	Standard PDF417
EDEV_OC_SYMBOL_PDF417_TRUNCATED	Truncated PDF417
EDEV_OC_SYMBOL_QRCODE_MODEL_1	QR Code Model 1
EDEV_OC_SYMBOL_QRCODE_MODEL_2	QR Code Model 2
EDEV_OC_SYMBOL_QRCODE_MICRO *	Micro QR Code
EDEV_OC_SYMBOL_MAXICODE_MODE_2	MaxiCode Mode 2
EDEV_OC_SYMBOL_MAXICODE_MODE_3	MaxiCode Mode 3
EDEV_OC_SYMBOL_MAXICODE_MODE_4	MaxiCode Mode 4
EDEV_OC_SYMBOL_MAXICODE_MODE_5	MaxiCode Mode 5
EDEV_OC_SYMBOL_MAXICODE_MODE_6	MaxiCode Mode 6
EDEV_OC_SYMBOL_GS1_DATABAR_STACKED	GS1 DataBar Stacked
EDEV_OC_SYMBOL_GS1_DATABAR_STACKED_ OMNIDIRECTIONAL	GS1 DataBar Stacked Omnidirectional
EDEV_OC_SYMBOL_GS1_DATABAR_EXPANDED_ STACKED	GS1 DataBar Expanded Stacked
EDEV_OC_SYMBOL_AZTECCODE_FULLRANGE	Aztec Code Full-Range モード
EDEV_OC_SYMBOL_AZTECCODE_COMPACT	Aztec Code Compact モード
EDEV_OC_SYMBOL_DATAMATRIX_SQUARE	DataMatrix ECC200 正方形
EDEV_OC_SYMBOL_DATAMATRIX_RECTANGLE_ 8	DataMatrix ECC200 長方形 行数 8
EDEV_OC_SYMBOL_DATAMATRIX_RECTANGLE_ 12	DataMatrix ECC200 長方形 行数 12
EDEV_OC_SYMBOL_DATAMATRIX_RECTANGLE_ 16	DataMatrix ECC200 長方形 行数 16

* TM-i ファームウェア Ver.4.1 以降対応

- level：　エラー訂正レベルを指定します。

設定値	説明
EDEV_OC_LEVEL_0	PDF417 エラー訂正レベル 0
EDEV_OC_LEVEL_1	PDF417 エラー訂正レベル 1
EDEV_OC_LEVEL_2	PDF417 エラー訂正レベル 2
EDEV_OC_LEVEL_3	PDF417 エラー訂正レベル 3
EDEV_OC_LEVEL_4	PDF417 エラー訂正レベル 4
EDEV_OC_LEVEL_5	PDF417 エラー訂正レベル 5
EDEV_OC_LEVEL_6	PDF417 エラー訂正レベル 6
EDEV_OC_LEVEL_7	PDF417 エラー訂正レベル 7
EDEV_OC_LEVEL_8	PDF417 エラー訂正レベル 8
EDEV_OC_LEVEL_L	QR Code エラー訂正レベル L
EDEV_OC_LEVEL_M	QR Code エラー訂正レベル M
EDEV_OC_LEVEL_Q	QR Code エラー訂正レベル Q
EDEV_OC_LEVEL_H	QR Code エラー訂正レベル H
整数値 (5 ～ 95)	Aztec Code エラー訂正レベル (デフォルト : 23)
EDEV_OC_LEVEL_DEFAULT	既定レベル

参考

- 2 次元シンボルの種類に合わせて選択してください。
- MaxiCode, 2 次元 GS1 DataBar の場合、EDEV_OC_LEVEL_DEFAULT を選択してください。
- Micro QR Code は、EDEV_OC_LEVEL_H をサポートしていません。

- width：　1 モジュールの幅を指定します。0 ～ 255 の整数値で指定します。

2 次元シンボルの種類	有効な値	デフォルト
PDF417	2 ～ 8	3
QR Code	3 ～ 16	3
MaxiCode	無視されます。	
2 次元 GS1 DataBar	2 ～ 8	2
Aztec Code	2 ～ 16	3
DataMatrix	2 ～ 16	3

- height：　1 モジュールの高さを指定します。0 ～ 255 の整数値で指定します。

2 次元シンボルの種類	有効な値	デフォルト
PDF417	2 ～ 8 (width に対する倍率)	3
QR Code	無視されます。	
MaxiCode		
2 次元 GS1 DataBar		
Aztec Code		
DataMatrix		

- size：　2 次元シンボルの最大サイズを指定します。0 ～ 65535 の整数値で指定します。

2 次元シンボルの種類	デフォルト	説明
PDF417	0(自動)	1 段あたりのコードワード数を指定
QR Code	無視されます。	
MaxiCode	無視されます。	
2 次元 GS1 DataBar		
Expanded Stacked	0(自動)	バーコードの最大幅を指定 (106 以上)
その他	無視されます。	
Aztec Code	無視されます。	
DataMatrix	無視されます。	

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addHLine

横罫線印字を命令バッファーに追加します。
横罫線の描画をします。

注意	ページモード では使用できません。

構文

```
- (int) addHLine:(long)x1 X2:(long)x2 Style:(int)style;
```

パラメーター

- x 1:　横罫線の描画開始位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- x2 :　横罫線の描画終了位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- style :　罫線の種類を指定します。

設定値	説明
EDEV_OC_LINE_THIN	実線 : 細
EDEV_OC_LINE_MEDIUM	実線 : 中太
EDEV_OC_LINE_THICK	実線 : 太
EDEV_OC_LINE_THIN_DOUBLE	二重線 : 細
EDEV_OC_LINE_MEDIUM_DOUBLE	二重線 : 中太
EDEV_OC_LINE_THICK_DOUBLE	二重線 : 太
EDEV_OC_PARAM_DEFAULT	実線 : 細

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addVLineBegin

縦罫線の開始を命令バッファーに追加します。縦罫線の描画を開始します。

注意	ページモードでは使用できません。
参考	縦罫線は、addVLineEnd (89 ページ)で終了を指定するまで描画します。本 API は addVLineEnd と一緒にお使いください。

構文

```
- (int) addVLineBegin:(long)x Style:(int)style;
```

パラメーター

- x：　縦罫線の描画開始位置(ドット単位)を指定します。0～65535 の整数値で指定します。
- style：　罫線の種類を指定します。

設定値	説明
EDEV_OC_LINE_THIN	実線：細
EDEV_OC_LINE_MEDIUM	実線：中太
EDEV_OC_LINE_THICK	実線：太
EDEV_OC_LINE_THIN_DOUBLE	二重線：細
EDEV_OC_LINE_MEDIUM_DOUBLE	二重線：中太
EDEV_OC_LINE_THICK_DOUBLE	二重線：太
EDEV_OC_PARAM_DEFAULT	実線：細

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addVLineEnd

縦罫線の終了を命令バッファーに追加します。縦罫線の描画を終了します。

注意	ページモード では使用できません。
参考	本 API は addVLineBegin (88 ページ) と一緒にお使いください。

構文

```
- (int) addVLineEnd:(long)x Style:(int)style;
```

パラメーター

- x :　縦罫線の描画を終了する位置(ドット単位)を指定します。0〜65535の整数値で指定します。
- style :　描画を終了する罫線の種類を指定します。

設定値	説明
EDEV_OC_LINE_THIN	実線 : 細
EDEV_OC_LINE_MEDIUM	実線 : 中太
EDEV_OC_LINE_THICK	実線 : 太
EDEV_OC_LINE_THIN_DOUBLE	二重線 : 細
EDEV_OC_LINE_MEDIUM_DOUBLE	二重線 : 中太
EDEV_OC_LINE_THICK_DOUBLE	二重線 : 太
EDEV_OC_PARAM_DEFAULT	実線 : 細

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPageBegin

ページモード開始を命令バッファーに追加します。ページモードの処理が開始します。

注意	ページモードで addImage (75 ページ)、addLogo (77 ページ) を使用する場合、多階調印刷はサポートしていません。
参考	縦罫線は、addPageEnd (90 ページ) で終了を指定するまでページモードで処理します。 本 API は PageEnd と一緒にお使いください。

構文

```
- (int) addPageBegin;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPageEnd

ページモード終了を命令バッファーに追加します。ページモードの処理が終了します。

参考	本 API は addPageBegin (90 ページ) と一緒にお使いください。

構文

```
- (int) addPageEnd;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPageArea

ページモード印字領域を命令バッファーに追加します。

ページモード印字領域(座標)を指定します。本APIに続けて、addTextメソッドなど印刷データのAPIを指定します。

注意	印字内容に合わせて印字領域を指定してください。印字データが印字領域をはみ出た場合、印字データが途中で切れた印字結果になります。
参考	本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (int) addPageArea:(long)x Y:(long)y
        Width:(long)width Height:(long)height;
```

パラメーター

- x：　横方向の原点 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
  0 はプリンターの印字可能領域の左端になります。
- y：　縦方向の原点 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
  0 は紙送りをしていない位置です。
- width：　印字領域の幅 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- height：　印字領域の高さ (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。

注意	印字領域の幅と高さは、印字方向の設定に合わせて確定してください。印字データが切れてしまう場合があります。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

4

addPageDirection

ページモード印字方向設定を命令バッファーに追加します。ページモードの印字方向を指定します。

> **参考**
> 本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (int) addPageDirection:(int)dir;
```

パラメーター

- dir :　ページモードの印字方向を指定します。

設定値	説明
EDEV_OC_DIRECTION_LEFT_TO_RIGHT (デフォルト)	回転しない (左上を始点に右方向へ印字)
EDEV_OC_DIRECTION_BOTTOM_TO_TOP	反時計回り 90 度回転 (左下を始点に上方向へ印字)
EDEV_OC_DIRECTION_RIGHT_TO_LEFT	180 度回転 (右下を始点に左方向へ印字)
EDEV_OC_DIRECTION_TOP_TO_BOTTOM	時計回り 90 度回転 (右上を始点に下方向へ印字)

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPagePosition

ページモードの印字位置設定領域を命令バッファーに追加します。
addPageArea (91 ページ) で指定したエリア内での、印字開始位置 (座標) を指定します。

参考　本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (int) addPagePosition:(long)x Y:(long)y;
```

パラメーター

- x：　横方向の印字位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- y：　縦方向の印字位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。

参考　印字開始位置 (座標) は、印字内容に合わせて指定してください。以下を参考にしてください。
* 文字列を印字する場合
  最初の文字のベースライン左端を指定します。
  標準の大きさで左詰めに印字する場合は省略可能です。高さが 2 倍の文字を印刷する場合は、y を 42 以上に指定します。
* バーコードを印字する場合
  シンボルの左下を指定します。y にバーコードの高さを指定してください。
* グラフィック / ロゴを印字する場合
  グラフィックデータの左下を指定します。y にグラフィックデータの高さを指定してください。
* 2 次元シンボルを印字する場合
  シンボルの左上を指定します。左上から印字する場合は、省略可能です。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPageLine

ページモードの直線描画を命令バッファーに追加します。ページモードで直線を描画します。

注意	斜線は描画できません。
参考	本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (int) addPageLine:(long)x1 Y1:(long)y1 X2:(long)x2
                    Y2:(long)y2 Style:(int)style;
```

パラメーター

- x 1:　横方向の描画開始位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- y 1:　縦方向の描画開始位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- x2 :　横方向の描画終了位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- y2 :　縦方向の描画終了位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- style :　罫線の種類を指定します。

設定値	説明
EDEV_OC_LINE_THIN	実線 : 細
EDEV_OC_LINE_MEDIUM	実線 : 中太
EDEV_OC_LINE_THICK	実線 : 太
EDEV_OC_LINE_THIN_DOUBLE	二重線 : 細
EDEV_OC_LINE_MEDIUM_DOUBLE	二重線 : 中太
EDEV_OC_LINE_THICK_DOUBLE	二重線 : 太
EDEV_OC_PARAM_DEFAULT	実線 : 細

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPageRectangle

ページモードの四角形描画を命令バッファーに追加します。ページモードで四角形を描画します。

参考

本 API は addPageBegin (90 ページ) と addPageEnd (90 ページ) に挟んでお使いください。

構文

```
- (int) addPageRectangle:(long)x1 Y1:(long)y1
                         X2:(long)x2 Y2:(long)y2
                         Style:(int)style;
```

パラメーター

- x 1:　横方向の描画開始位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- y 1:　縦方向の描画開始位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- x2 :　横方向の描画終了位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- y2 :　縦方向の描画終了位置 (ドット単位) を指定します。0 ～ 65535 の整数値で指定します。
- style :　線の種類を指定します。

設定値	説明
EDEV_OC_LINE_THIN	実線 : 細
EDEV_OC_LINE_MEDIUM	実線 : 中太
EDEV_OC_LINE_THICK	実線 : 太
EDEV_OC_LINE_THIN_DOUBLE	二重線 : 細
EDEV_OC_LINE_MEDIUM_DOUBLE	二重線 : 中太
EDEV_OC_LINE_THICK_DOUBLE	二重線 : 太
EDEV_OC_PARAM_DEFAULT	実線 : 細

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addCut

用紙カットを命令バッファーに追加します。用紙カットを設定します。

> **注意**
> ページモードでは使用できません。

構文

```
- (int) addCut:(int)type;
```

パラメーター

- type：　　用紙カット方法を指定します。

設定値	説明
EDEV_OC_CUT_NO_FEED	フィードなしカット （紙送りせずにカット）
EDEV_OC_CUT_FEED	フィードカット （紙送り後カット）
EDEV_OC_CUT_RESERVE	カット予約 （後に続く印字を実行後、カット位置でカット）
EDEV_OC_PARAM_DEFAULT	フィードカット （紙送り後カット）

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addPulse

ドロアーキックを命令バッファーに追加します。ドロアーキックを設定します。

注意
- ページモードでは使用できません。
- ドロアーは、ブザーと一緒に使用できません。

構文

```
- (int) addPulse:(int)drawer Time:(int)time;
```

パラメーター

- drawer：　ドロアーキックコネクターを指定します。

設定値	説明
EDEV_OC_DRAWER_1	ドロアーキックコネクター 2 番ピン
EDEV_OC_DRAWER_2	ドロアーキックコネクター 5 番ピン
EDEV_OC_PARAM_DEFAULT	ドロアーキックコネクター 2 番ピン

- time：　ドロアーキック信号の通電時間を指定します。

設定値	説明
EDEV_OC_PULSE_100	100 ミリ秒の信号
EDEV_OC_PULSE_200	200 ミリ秒の信号
EDEV_OC_PULSE_300	300 ミリ秒の信号
EDEV_OC_PULSE_400	400 ミリ秒の信号
EDEV_OC_PULSE_500	500 ミリ秒の信号
EDEV_OC_PARAM_DEFAULT	100 ミリ秒の信号

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addSound

ブザーの鳴動を命令バッファーに追加します。ブザーを設定します。

注意
- ページモードでは使用できません。
- ブザーの機能は、ドロアーと一緒に使用できません。
- 本 API はプリンターにブザーが付いてなければ使用できません。

構文

```
- (int) addSound:(int)pattern Repeat:(long)repeat
                 Cycle:(long)cycle;
```

パラメーター

- pattern : ブザーの音色を指定します。

設定値	説明
EDEV_OC_PATTERN_NONE	停止
EDEV_OC_PATTERN_A	パターン A (外付けオプションブザー)
EDEV_OC_PATTERN_B	パターン B (外付けオプションブザー)
EDEV_OC_PATTERN_C	パターン C (外付けオプションブザー)
EDEV_OC_PATTERN_D	パターン D (外付けオプションブザー)
EDEV_OC_PATTERN_E	パターン E (外付けオプションブザー)
EDEV_OC_PATTERN_ERROR	エラー鳴動パターン (外付けオプションブザー)
EDEV_OC_PATTERN_PAPER_END	用紙なし鳴動パターン (外付けオプションブザー)
EDEV_OC_PATERN_1	パターン 1 (内蔵ブザー)
EDEV_OC_PATERN_2	パターン 2 (内蔵ブザー)
EDEV_OC_PATERN_3	パターン 3 (内蔵ブザー)
EDEV_OC_PATERN_4	パターン 4 (内蔵ブザー)
EDEV_OC_PATERN_5	パターン 5 (内蔵ブザー)
EDEV_OC_PATERN_6	パターン 6 (内蔵ブザー)
EDEV_OC_PATERN_7	パターン 7 (内蔵ブザー)
EDEV_OC_PATERN_8	パターン 8 (内蔵ブザー)
EDEV_OC_PATERN_9	パターン 9 (内蔵ブザー)
EDEV_OC_PATERN_10	パターン 10 (内蔵ブザー)
EDEV_OC_PARAM_DEFAULT	パターン A

- repeat : 繰り返し回数を指定します。

設定値	説明
0	無限
1 ～ 255	1 ～ 255 回
EDEV_OC_PARAM_DEFAULT	1 回

参考
repeat に "0" を指定後、ブザーを停止したい場合、本 API を再度実行し、pattern に EDEV_OC_PATTERN_NONE を指定してください。

- cycle：　ブザーを鳴らす周期をミリ秒単位で指定します。有効な値の範囲は 1000 ～ 25500 です。

ブザーの音色 EDEV_OC_PATTERN_1 ～ EDEV_OC_PATTERN_10 で有効です。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addLayout

用紙レイアウトの設定を命令バッファーに追加します。

注意	用紙レイアウトの設定はスタンダードモードで行ってください。ページモードでは、本 API の設定は無効です。

構文

```
- (int) addLayout:(int)type Width:(long)width
                  Height:(long)height
                  MarginTop:(long)marginTop
                  MarginBottom:(long)marginBottom
                  OffsetCut:(long)offsetCut
                  OffsetLabel:(long)offsetLabel;
```

パラメーター

- type : 用紙の種類を指定します。

設定値	説明
EDEV_OC_LAYOUT_RECEIPT	レシート
EDEV_OC_LAYOUT_RECEIPT_BM	レシート (ブラックマークあり)
EDEV_OC_LAYOUT_LABEL	ダイカットラベル紙
EDEV_OC_LAYOUT_LABEL_BM	ダイカットラベル紙 (ブラックマークあり)

- width : 用紙の幅 (0.1 mm 単位) を指定します。有効な値の範囲は 290 ～ 600 です。
- height : 用紙の高さ (0.1 mm 単位) を指定します。

用紙種類	有効な値	説明
レシート	0	指定不要
レシート (BM あり)	• 0 (自動) • 284 ～ 1550 (手動)	BM 上端位置から、次の BM 上端位置までの距離
ダイカットラベル紙		ラベル上端位置から、次のラベル上端位置までの距離
ダイカットラベル紙 (BM あり)		BM 下端位置から、次の BM 下端位置までの距離

- margin_top : 上余白 (0.1 mm 単位) を指定します。

用紙種類	有効な値	説明
レシート	0	指定不要
レシート (BM あり)	-150 ～ 1500	BM 上端位置からの距離
ダイカットラベル紙	0 ～ 1500	ラベル上端位置からの距離
ダイカットラベル紙 (BM あり)	-15 ～ 1500	BM 下端位置からの距離

- margin_bottom :(
  下余白 (0.1 mm 単位) を指定します。

用紙種類	有効な値	説明
レシート	0	指定不要
レシート (BM あり)	0	指定不要
ダイカットラベル紙	-15 ～ 0	ラベル下端位置からの距離 (紙送り方向が正の数)
ダイカットラベル紙 (BM あり)	-15 ～ 15	BM 上端位置からの距離 (紙送り方向が正の数)

- offset_cut :　カット位置 (0.1 mm 単位) を指定します。

用紙種類	有効な値	説明
レシート	0	指定不要
レシート (BM あり)	-290 ～ 50	BM 上端位置から、カット位置までの距離
ダイカットラベル紙	0 ～ 50	ラベル下端位置から、カット位置までの距離
ダイカットラベル紙 (BM あり)	0 ～ 50	BM 上端位置から、カット位置までの距離

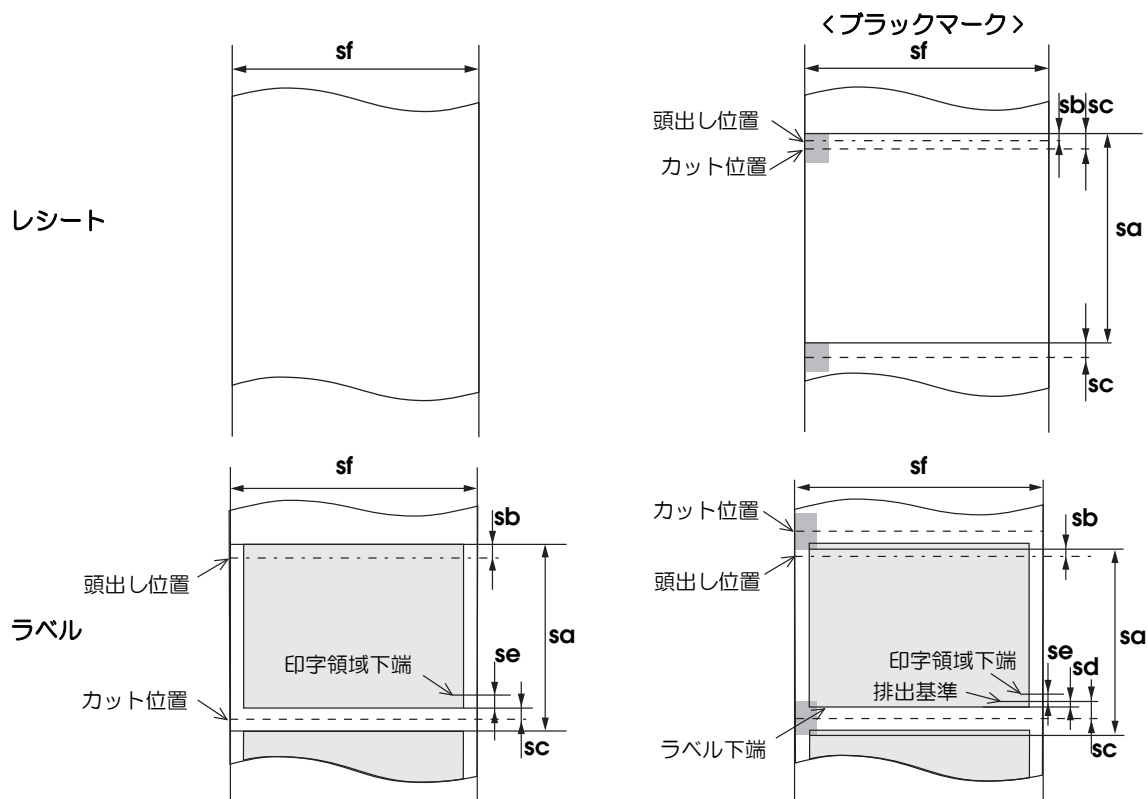
- offset_label :　ラベル下端位置 (0.1 mm 単位) を指定します。

用紙種類	有効な値	説明
レシート	0	指定不要
レシート (BM あり)	0	指定不要
ダイカットラベル紙	0	指定不要
ダイカットラベル紙 (BM あり)	0 ～ 15	BM 上端位置から、ラベル下端位置までの距離

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

詳細説明

用紙ごと指定可能なパラメーターの位置は、以下を参照してください。

記号	パラメーター
sf	width
sa	height
sb	margin_top
se	margin_bottom
sc	offset_cut
sd	offset_label

addCommand

コマンドを命令バッファーに追加します。ESC/POS コマンドを送信します。

ESC/POS コマンドの詳細については ESC/POS アプリケーションプログラミングガイドを参照してください。ESC/POS アプリケーションプログラミングガイドの入手には弊社との機密保持契約が必要です。詳しくは販売元 までお問い合わせください。

構文

```
- (int) addCommand:(NSData *)data;
```

パラメーター

- data：　ESC/POS コマンドをバイナリーデータで指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addRecovery

エラーからの復帰を命令バッファーに追加します。

本 API は、自動復帰できないエラー発生時、エラー要因除去後に実行します。

- 復帰可能エラーから復帰し、プリンターのバッファーをクリアします。
  本 API は強制送信モードを有効にして使用してください。
- 本機能は、recover (106 ページ) を使用することを推奨します。

構文

```
- (int) addRecovery;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

addReset

プリンターをリセットします。

- 印刷ドキュメントに含まれている他の印字命令は実行されません。
- 本機能は、reset (106 ページ) を使用することを推奨します。

構文

```
- (int) addReset;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

startMonitor

プリンターのステータスイベントの通知を開始します。

構文

```
- (int) startMonitor;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

stopMonitor

プリンターのステータスイベント通知を終了します。

構文

```
- (int)stopMonitor;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

recover

プリンターをエラー状態から復帰します。

本 API は、自動復帰可能なエラー発生時に実行します。

> **参考**
> 本 API は force プロパティ (109 ページ) の設定に関係なく、エラー状態から復帰します。

構文

```
- (int) recover;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理が実行できなかった。

reset

プリンターをリセットします。

> **参考**
> 本 API を実行すると、force プロパティ (109 ページ) の設定が、EDEV_OC_FALSE になります。

構文

```
- (int) reset;
```

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理が実行できなかった。

halftone プロパティ

ハーフトーン処理方法を設定します。
addImage（75 ページ）のモノクロ（2 階調）印刷に適用されます。

構文

Setter

```
-(int) setHalftone:(int) halftone
```

Getter

```
-(int) getHalftone
```

設定値

設定値	説明
EDEV_OC_HALFTONE_DITHER(デフォルト)	ディザ、グラフィックのみの印刷に適しています。
EDEV_OC_HALFTONE_ERROR_DIFFUSION	誤差拡散、文字とグラフィックが混在する印刷に適しています。
EDEV_OC_HALFTONE_THRESHOLD	しきい値、文字のみの印刷に適しています。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。

brightness プロパティ

明るさ補正値を設定します。

addImage (75 ページ)の実行時に適用されます。

構文

Setter

```
-(int) setBrightness:(double) brightness
```

Getter

```
-(double) getBrightness
```

設定値

明るさ補正値をガンマー値 0.1 ～ 10.0 の範囲で指定します。

デフォルトは 1.0 です。

参考	明るさ補正値に 1.0 以外を指定した場合、印字速度が遅くなります。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。

force プロパティ

強制送信モードを設定します。

強制送信モードを有効にすると、プリンターへ強制的に印字命令を送信します。

> **参考**
> - 強制送信モードはプリンターがオフラインの状態で使用してください。
>   プリンターがオンライン状態で使用した場合、応答ドキュメント受信イベント（111 ページ）の code に EDEV_OC_ERR_PORT が返ります。
> - 強制送信モードで有効な機能は以下のとおりです。
>   * ドロアーキック（addPulse（97 ページ））
>   * ブザーの停止（addSound（98 ページ））
>   * 復帰可能エラーからの復帰（addRecovery（104 ページ））
>   * リセット（addReset（104 ページ））
>   * リアルタイムコマンドの送信（addCommand（103 ページ））

構文

Setter

```
-(int) setForce:(BOOL) force
```

Getter

```
-(BOOL) getForce
```

設定値

設定値	説明
YES	強制送信モードを有効にします。
NO	強制送信モードを無効にします。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。

timeout プロパティ

sendData (62 ページ) で送信するメッセージのタイムアウト (ms) を設定します。

構文

Setter

```
-(int) setTimeout:(long) timeout
```

Getter

```
-(long) getTimeout
```

設定値

タイムアウトを 0 ～ 600,000 の範囲で指定します。
デフォルトは 10,000 です。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。

interval プロパティ

ステータスを更新する間隔 (ms) を指定します。

構文

Setter

```
-(int) setInterval:(long) interval
```

Getter

```
-(long) getInterval
```

設定値

ステータスを更新する間隔を 1,000 ～ 6,000 の範囲で指定します。
デフォルトは 3,000 です。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。

setReceiveEventCallback

応答ドキュメント受信イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setReceiveEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

❏
```
- (void) メソッドの名称: (NSString *)ipAddress
              DeviceId:(NSString *)deviceId
              Success:(int)success
              Code:(int)code
              Status:(NSNumber *)status
              Battery:(NSNumber *)battery
```

❏ ジョブ ID 指定用
```
- (void) メソッドの名称: (NSString *)ipAddress
              DeviceId:(NSString *)deviceId
              Success:(int)success
              Code:(int)code
              Status:(NSNumber *)status
              Battery:(NSNumber *)battery
              PrintJobId:(NSString *)printJobId
```

> **注意**
> printJobId パラメーターが付くメソッドの使用は、TM-i ファームウェア Ver.4.1 以降で対応しています

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- success:　印刷の実行結果がセットされます。

エラーステータス	説明
EDEV_OC_TRUE	• 印刷成功 • 処理が成功 (スプーラー機能が有効時)
EDEV_OC_FALSE	• 印刷失敗 • 処理が失敗 (スプーラー機能が有効時)

- code:　処理の実行結果がセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_PRINTING *	印刷中。
EDEV_OC_ERR_AUTOMATICAL	自動復帰エラーが発生した。
EDEV_OC_ERR_BATTERY_LOW	バッテリー残量なし
EDEV_OC_ERR_COVER_OPEN	カバーオープンエラーが発生した。
EDEV_OC_ERR_CUTTER	オートカッターエラーが発生した。
EDEV_OC_ERR_MECHANICAL	メカニカルエラーが発生した。
EDEV_OC_ERR_EMPTY	ロール紙エンド検出器に用紙ない。
EDEV_OC_ERR_UNRECOVERABLE	復帰不可能エラーが発生した。
EDEV_OC_ERR_FAILURE	要求ドキュメントの構文に誤りがある。
EDEV_OC_ERR_NOT_FOUND	デバイス ID で指定したプリンターが存在しない
EDEV_OC_ERR_SYSTEM	印刷システムにエラーが発生した。
EDEV_OC_ERR_PORT	通信ポートに異常を検出した。
EDEV_OC_ERR_TIMEOUT	印刷タイムアウトが発生した。
EDEV_OC_ERR_JOB_NOT_FOUND *	指定のジョブ ID が存在しない。
EDEV_OC_ERR_SPOOLER *	印刷キューに空き容量がない。

* TM-i ファームウェア Ver.4.1 以降対応

- Status:　　プリンターステータスがセットされます。

プリンターステータス	説明
EDEV_OC_ST_NO_RESPONSE	TM プリンター無応答
EDEV_OC_ST_PRINT_SUCCESS	印刷終了 / スプール完了
EDEV_OC_ST_DRAWER_KICK	ドロアーキックコネクター 3 番ピンの状態＝ ″H″
EDEV_OC_ST_BATTERY_OFFLINE	バッテリー残量によるオフライン状態
EDEV_OC_ST_OFF_LINE	オフライン状態
EDEV_OC_ST_COVER_OPEN	カバーオープン
EDEV_OC_ST_PAPER_FEED	紙送りスイッチによる紙送り中
EDEV_OC_ST_WAIT_ON_LINE	オンライン復帰待ち中
EDEV_OC_ST_PANEL_SWITCH	紙送りスイッチが押下中
EDEV_OC_ST_MECHANICAL_ERR	メカニカルエラー発生
EDEV_OC_ST_AUTOCUTTER_ERR	オートカッターエラー発生
EDEV_OC_ST_UNRECOVER_ERR	復帰不可能エラー発生
EDEV_OC_ST_AUTORECOVER_ERR	自動復帰エラー発生
EDEV_OC_ST_RECEIPT_NEAR_END	ロール紙ニアエンド検出器に用紙なし
EDEV_OC_ST_RECEIPT_END	ロール紙エンド検出器に用紙なし
EDEV_OC_ST_BUZZER	ブザーが鳴動中 (対応機器のみ)
EDEV_OC_ST_SPOOLER_IS_STOPPED *	スプーラーの停止

* TM-i ファームウェア Ver.4.1 以降対応

- Battery:　　バッテリーステータスがセットされます。

電源の状態

バッテリーステータス	説明
0x30XX	AC アダプターが接続されている
0x31XX	AC アダプターが接続されていない

バッテリー残量

バッテリーステータス	説明
0xXX36	バッテリー残量 6
0xXX35	バッテリー残量 5
0xXX34	バッテリー残量 4
0xXX33	バッテリー残量 3
0xXX32	バッテリー残量 2
0xXX31	バッテリー残量 1(ニアエンド)
0xXX30	バッテリー残量 0(リアルエンド)

参考

バッテリー非搭載機種の場合は 0 になります。

- printJobId:　　印刷のジョブ ID がセットされます。

setStatusChangeEventCallback

ステータス変更イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setStatusChangeEventCallback: (SEL) callback
                               Target: (NSObject*) target;
```

パラメーター

- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                        DeviceId:(NSString *)deviceId
                        Status:(NSNumber *)status
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。
- status: イベント発生時のプリンターステータスがセットされます。

setBatteryStatusChangeEventCallback

バッテリーステータス変更イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ) 実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setBatteryStatusChangeEventCallback:
                                    (SEL) callback
                                    Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                    DeviceId:(NSString *)deviceId
                    Battery:(NSNumber *)battery
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- battery:　イベント発生時のバッテリーステータスがセットされます。

4

setOnlineEventCallback

オンラインイベントを受け取る、コールバックメソッドを登録します。

参考

- 本 API は、createDevice (41 ページ) 実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setOnlineEventCallback: (SEL) callback
                                  Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考

本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                      DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

setOfflineEventCallback

オフラインイベントを受け取る、コールバックメソッドを登録します。

参考
- 本 API は、createDevice (41 ページ) 実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setOfflineEventCallback: (SEL) callback
                                  Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考
本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                      DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

4

setPowerOffEventCallback

無応答イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ) 実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setPowerOffEventCallback: (SEL) callback
                            Target: (NSObject*) target;
```

パラメーター

- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                 DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。

setCoverOkEventCallback

カバークローズイベントを受け取る、コールバックメソッドを登録します。

参考
- 本 API は、createDevice (41 ページ)実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setCoverOkEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考
本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                    DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

setCoverOpenEventCallback

カバーオープンイベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setCoverOpenEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                  DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。

setPaperOkEventCallback

用紙ありイベントを受け取る、コールバックメソッドを登録します。

参考
- 本 API は、createDevice (41 ページ)実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setPaperOkEventCallback: (SEL) callback
                         Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考
本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

setPaperNearEndEventCallback

用紙残量少イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setPaperNearEndEventCallback: (SEL) callback
                              Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                       DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

setPaperEndEventCallback

用紙なしイベントを受け取る、コールバックメソッドを登録します。

参考

- 本 API は、createDevice (41 ページ)実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setPaperEndEventCallback: (SEL) callback
                         Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考

本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

4

setDrawerClosedEventCallback

ドロアークローズイベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setDrawerClosedEventCallback: (SEL) callback
                             Target: (NSObject*) target;
```

パラメーター

- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                   DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。

setDrawerOpenEventCallback

ドロアーオープンイベントを受け取る、コールバックメソッドを登録します。

参考
- 本 API は、createDevice (41 ページ)実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setDrawerOpenEventCallback: (SEL) callback
                              Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考
本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                   DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

setBatteryLowEventCallback

バッテリー残量なしイベントを受け取る、コールバックメソッドを登録します。

> **参考**
> - 本 API は、createDevice (41 ページ)実行後に使用してください。
> - 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setBatteryLowEventCallback: (SEL) callback
                         Target: (NSObject*) target;
```

パラメーター

- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                    DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress: TM-i の IP アドレスがセットされます。
- deviceId: デバイス ID がセットされます。

setBatteryOkEventCallback

バッテリー残量ありイベントを受け取る、コールバックメソッドを登録します。

参考
- 本 API は、createDevice (41 ページ)実行後に使用してください。
- 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setBatteryOkEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考
本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                    DeviceId:(NSString *)deviceId
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。

EposScanner クラス

setDataEventCallback

バーコード情報の読み取りイベントを受け取る、コールバックメソッドを登録します。

参考	本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setDataEventCallback: (SEL) callback
                       Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

参考	本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                 DeviceId:(NSString *)deviceId
                 Input:(NSString *)input;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- deviceId:　デバイス ID がセットされます。
- input:　検出された文字列がセットされます。

EposSimpleSerial クラス

sendCommand

任意コマンドを送信します。

構文

```
- (int)sendCommand:(NSData *)data;
```

パラメーター

- data:　コマンドを 16 進数文字列で指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

setCommandReplyEventCallback

デバイスからの応答イベントを受け取る、コールバックメソッドを登録します。

> **参考**
> 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setCommandReplyEventCallback: (SEL) callback
                           Target: (NSObject*) target;
```

パラメーター

- callback:　　コールバックメソッドのセレクターを指定します。
- target:　　コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                DeviceId:(NSString *)deviceId
                Data:(NSData *)data;
```

パラメーター

- ipAddress:　　TM-i の IP アドレスがセットされます。
- deviceId:　　デバイス ID がセットされます。
- data:　　応答データのバイナリーデータがセットされます。

EposCommBoxManager クラス

openCommBox

コミュニケーションボックスをオープンします。

構文

```
- (int)openCommBox:(NSString *)boxId
                   MemberId:(NSString *)memberId
                   Sequence:(long *)sequence
                   Callback:(SEL)callback
                   Target:(NSObject*) target;
```

パラメーター

- boxId:　オープンするボックス ID を指定します。
- memberId:　コミュニケーションボックス内で、自身を識別するためのメンバー ID を指定します。
- sequence:　本 API の実行時に処理のシーケンス番号がセットされます。
- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	通信できなかった。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
         CommBoxObject:(EposCommBox *)commBoxObject
         Code:(int)code Sequence:(long)sequence;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- commBoxObject:
  EposCommBox オブジェクトがセットされます。
  本 API の処理に失敗した場合、nil がセットされます。

- code:　　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	コミュニケーションボックスをオープンに成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_ALREADY_OPENED	すでにコミュニケーションボックスがオープンしている。
EDEV_OC_ERR_ALREADY_USED	指定したメンバー ID はすでに使用されている。
EDEV_OC_ERR_BOX_COUNT_OVER	コミュニケーションボックスを作成可能な最大数を超えている。
EDEV_OC_ERR_BOX_CLIENT_OVER	コミュニケーションボックスに所属可能な最大数を超えている。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

- sequence:　本 API の実行時に、第 3 パラメーターに返された値と同じシーケンス番号がセットされます。

closeCommBox

コミュニケーションボックスをクローズします。

構文

```
- (int)closeCommBox:(EposCommBox *)commBoxObject
                    Sequence:(long *)sequence
                    Callback:(SEL)callback
                    Target:(NSObject*) target;
```

パラメーター

- commBoxObject:
  クローズ対象の EposCommBox オブジェクトを指定します。
- sequence:　本 API の実行時に処理のシーケンス番号がセットされます。
- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	通信できなかった。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
         Code:(int)code Sequence:(long)sequence;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- code:　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	コミュニケーションボックスをオープンに成功した。
EDEV_OC_ERR_CONNECT	デバイスとの通信に失敗した。
EDEV_OC_ERR_NOT_OPENED	コミュニケーションボックスが開いていない。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

- sequence:　本 API の実行時に、第 2 パラメーターに返された値と同じシーケンス番号がセットされます。

4

EposCommBox クラス

getCommHistory

コミュニケーションボックスの送信履歴を取得します。

構文

```
- (int)getCommHistory: (long *)sequence
                Callback:(SEL)callback
                Target:(NSObject*) target;
```

パラメーター

- sequence:　本 API の実行時に処理のシーケンス番号がセットされます。
- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	通信できなかった。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                    Code:(int)code
                    HistoryList:(NSArray *)historyList
                    Sequence:(long)sequence;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- code:　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_NOT_OPENED	コミュニケーションボックスが開いていない。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

- historyList:　コミュニケーションボックスの通信履歴がセットされます。

(NSDictionary *)historyList[index]	
キー	説明
(NSString *)senderId	送信元の memberID
(NSString *)receiveId	送信先の memberID
(NSString *)message	受信したメッセージ

- sequence:　本 API の実行時に、第 1 パラメーターに返された値と同じシーケンス番号がセットされます。

4

sendData

メッセージをコミュニケーションボックスに送信します。

構文

```
- (int)sendData:(NSString *)message
               MemberId:(NSString *)memberId
               Sequence:(long *)sequence
               Callback:(SEL)callback
               Target:(NSObject*) target;
```

パラメーター

- message: コミュニケーションボックスに送信するデータを指定します。
- memberId: 送信先のメンバー ID を指定します。
  nil または ” ”（空文字）を指定した場合、送信者を除くコミュニケーションボックス内のすべてのメンバーに送信します。
- sequence: 本 API の実行時に処理のシーケンス番号がセットされます。
- callback: コールバックメソッドのセレクターを指定します。
- target: コールバックメソッドを持つオブジェクトを指定します。

戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_CONNECT	通信できなかった。
EDEV_OC_ERR_MEMORY	処理に必要なメモリーが確保できなかった。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。
EDEV_OC_ERR_PROCESSING	処理を実行できなかった。

コールバックメソッドの定義

```
- (void) メソッドの名称 : (NSString *)ipAddress
                    Code:(int)code Count:(long)count
                    Sequence:(long)sequence;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- code:　処理結果のエラーステータスがセットされます。

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_NOT_OPENED	コミュニケーションボックスが開いていない。
EDEV_OC_ERR_NOT_FOUND	指定したメンバー ID が見つからない。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

- count:　メッセージの送信を試みたクライアント数がセットされます。
- sequence:　本 API の実行時に、第 3 パラメーターに返された値と同じシーケンス番号がセットされます。

setReceiveEventCallback

コミュニケーションボックスのメッセージ受信を通知する、コールバックメソッドを登録します。

> **参考**
> 本 API を複数回実行した場合、後に指定されたコールバックメソッドに上書きされます。

構文

```
- (void) setReceiveEventCallback: (SEL) callback
                          Target: (NSObject*) target;
```

パラメーター

- callback:　コールバックメソッドのセレクターを指定します。
- target:　コールバックメソッドを持つオブジェクトを指定します。

> **参考**
> 本 API 実行時に、callback と target のいずれかに nil を指定した場合、コールバックメソッドの登録は解除されます。

コールバックメソッドの定義

```
- (void) メソッドの名称: (NSString *)ipAddress
                 SenderId:(NSString *)senderId
                 ReceiverId:(NSString *)receiverId
                 Message:(NSString *)message;
```

パラメーター

- ipAddress:　TM-i の IP アドレスがセットされます。
- senderId:　送信元のメンバー ID がセットされます。
  openCommBox 実行時に、メンバー ID の指定がない場合、" "(空文字)がセットされます。
- receiverId:　送信先のメンバー ID がセットされます。
  コミュニケーションボックス内のすべてのメンバーに送信した場合、" "(空文字)がセットされます。
- message:　受信したデータがセットされます。

EposDeviceLog クラス

setLogSettings

ログ出力機能を設定します。

構文

```
+ (int) setLogSettings:(int) period
                   Enabled:(int) enabled
                   IpAddress:(NSString *)ipAddress
                   Port:(int)port LogSize:(int)logSize
                   LogLevel:(int)logLevel;
```

パラメーター

- period:　ログ出力機能の設定方法を指定します。

設定値	説明
EDEV_OC_LOG_TEMPORARY	アプリケーションを終了すると、本 API の設定は無効になります。
EDEV_OC_LOG_PERMANENT	アプリケーションを終了させても、本 API の設定を有効にします。

- enabled:　ログ出力機能の有効 / 無効、およびログの出力先を指定します。

設定値	説明
EDEV_OC_LOG_DISABLE	ログ出力機能を無効にします。
EDEV_OC_LOG_STORAGE	端末のストレージに出力します。
EDEV_OC_LOG_TCP	TCP で出力します。

参考

enabled を EDEV_OC_LOG_STORAGE に指定する場合、iTunes のファイル共有を可能にしてください。以下の手順で設定します。

1. アプリケーションの info.plist に "UIFileSharingEnabled" を追加します。
   自動で "Application supports iTunes file sharing" に変更されます。
2. "Application supports iTunes file sharing" の Value を、"YES" に設定します。

- ipAddress:　TCP 通信の IP アドレス (IPv4 形式) を指定します。
  1 ～ 50(MB 単位) の整数値を指定します。

参考

enabled の設定が以下の場合、nil も指定できます。
＊ EDEV_OC_LOG_DISABLE
＊ EDEV_OC_LOG_STORAGE

- port:　TCP 通信のポート番号を指定します。0 ～ 65,535 の整数値を指定します。

参考

enabled の設定が以下の場合も、範囲内の任意の値を指定してください。
＊ EDEV_OC_LOG_DISABLE
＊ EDEV_OC_LOG_STORAGE

4

- logSize:　端末のストレージへ保存する、ログの最大容量を指定します。

> **参考**
> enabled の設定が以下の場合も、範囲内の任意の値を指定してください。
> * EDEV_OC_LOG_DISABLE
> * EDEV_OC_LOG_TCP

- logLevel:　ログの出力レベルを指定します。

設定値	説明
EDEV_OC_LOG_LOW	低レベル

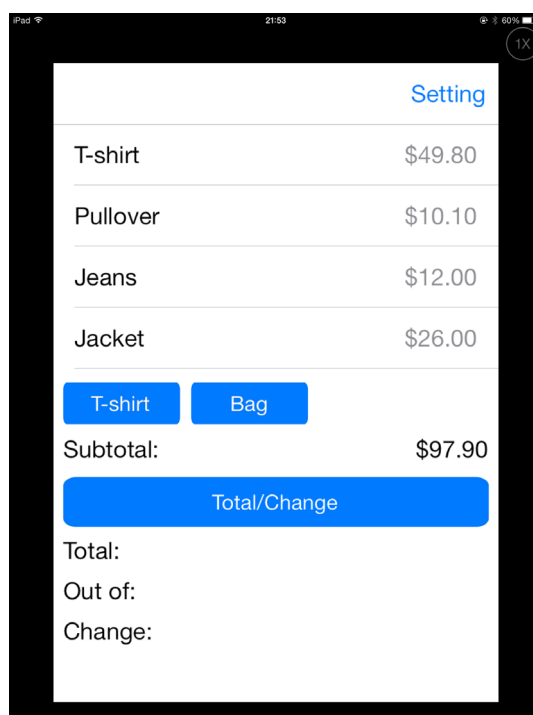
戻り値

エラーステータス	説明
EDEV_OC_SUCCESS	処理に成功した。
EDEV_OC_ERR_PARAM	不正なパラメーターが渡された。
EDEV_OC_ERR_FAILURE	その他のエラーが発生した。

サンプルプログラム

本章では、サンプルプログラム (ePOS-Device Sample Program for iOS) の使い方について説明しています。

参考
- サンプルプログラムは iOS アプリケーション開発者向けの、ePOS-Device for iOS API を使用した iOS アプリケーションの実装サンプルとして提供します。
- サンプルプログラムのパッケージは、Objective-C ソースファイルを含む Xcode 用 iOS アプリケーションプロジェクトとして提供しています。

機能

サンプルプログラムは、以下の機能を実装しています。

❏ TM-i と接続 / 切断
❏ デバイスオブジェクトの生成
❏ バーコードの読み取り
❏ カスタマーディスプレイの表示
❏ レシートの印刷
❏ プリンターのステータス取得
❏ エラー処理

使用環境

開発環境

- Xcode Version 4.2 ～ 4.6
- Xcode Version 5.0

必要なデバイス

- TM-i
- カスタマーディスプレイ
- バーコードスキャナー

iOS デバイス

❏ iPhone 4/ iPhone 4s/ iPhone 5/ iPhone 5s/ iPhone 5c/ iPhone 6/ iPhone 6 Plus
❏ iPod touch (3rd generation)/ iPod touch (4th generation)/ iPod touch (5th generation)
❏ iPad/ iPad 2/ iPad (3rd generation)/ iPad (4th generation)/ iPad Air
❏ iPad mini/ iPad mini with Retina display

環境構築

サンプルプログラムの環境設定のフローを以下に示します。

参考　EPSON TMNet WebConfig にアクセスするさい、EPSON TMNet WebConfig のユーザー名とパスワードが要求されます。

1. LAN ケーブルの接続

TM-i に LAN ケーブルを接続します。

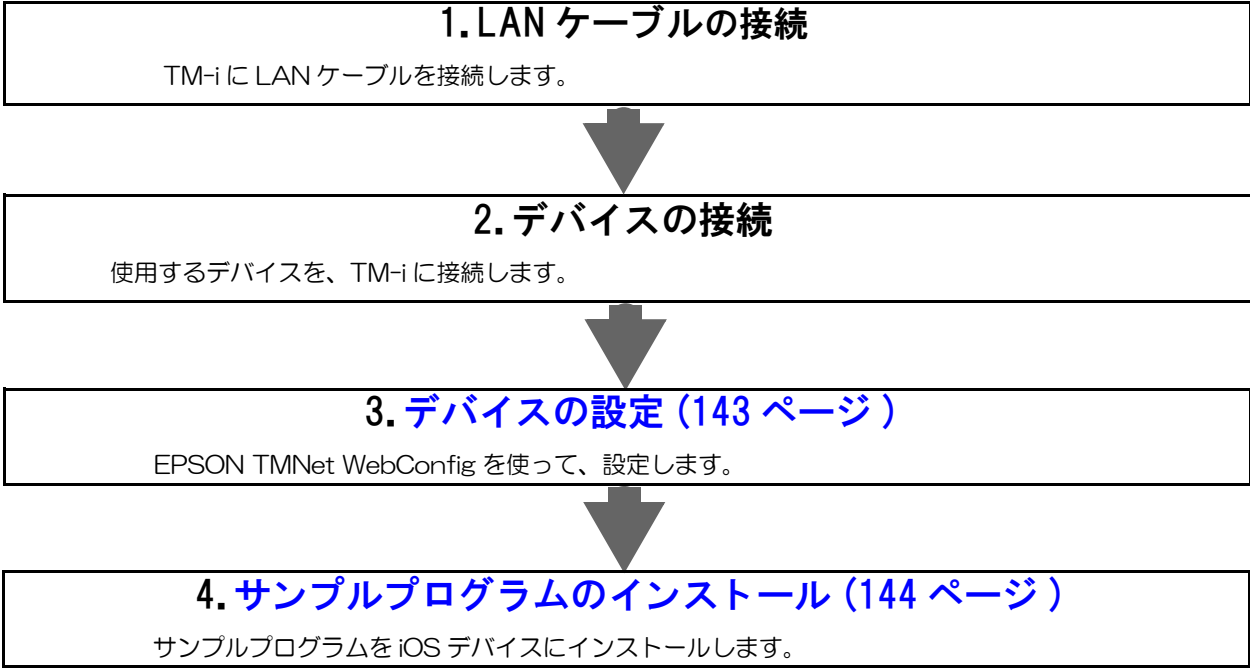
2. デバイスの接続

使用するデバイスを、TM-i に接続します。

3. デバイスの設定（143 ページ）

EPSON TMNet WebConfig を使って、設定します。

4. サンプルプログラムのインストール（144 ページ）

サンプルプログラムを iOS デバイスにインストールします。

デバイスの設定

Web ブラウザーから EPSON TMNet WebConfig を起動します。

参考　設定方法の詳細は、各 TM-i の詳細取扱説明書を参照してください。

サンプルプログラムを使用する環境には、以下のデバイスを設定する必要があります。

❏ カスタマーディスプレイ

設定項目	設定値
カスタマーディスプレイ	" 使用する "

参考　カスタマーディスプレイのデバイス ID は、"local_display" 固定です。

❏ バーコードスキャナー

設定項目	設定値
デバイス ID	local_scanner
デバイス名	リストから該当するデバイスを選択します。
制御スクリプト	Scanner_Generic.js

サンプルプログラムのインストール

以下の手順で iOS デバイスにサンプルプログラムをインストールします。

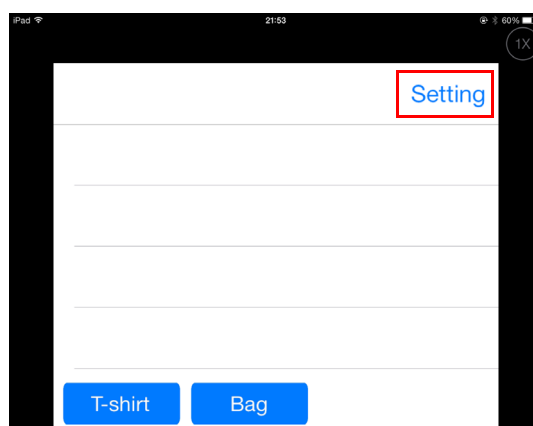
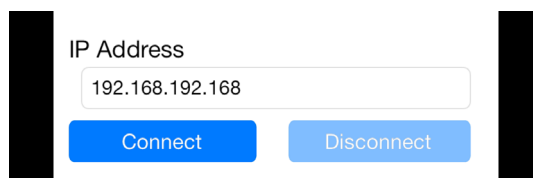
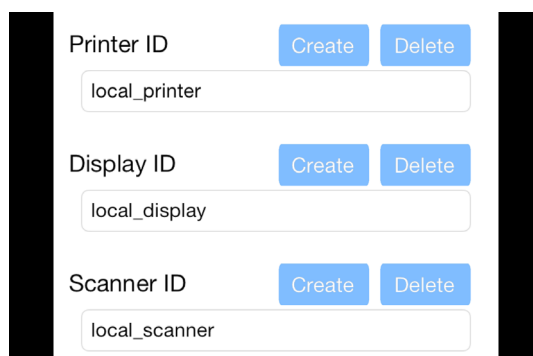
1. 開発環境のコンピューターに、ターゲットデバイスを USB で接続します。
2. サンプルプログラムの ZIP ファイルを、開発環境の任意のディレクトリーに展開します。
3. 展開したディレクトリー内に格納されている、"Composite_Sample.xcodeproj" をダブルクリックします。
4. Xcode が起動されます。「Scheme」としてターゲットデバイスを選択します。
5. 左上の [Run] ボタンをクリックします。
6. ターゲットデバイスにサンプルプログラムがインストールされ、サンプルプログラムが起動します。

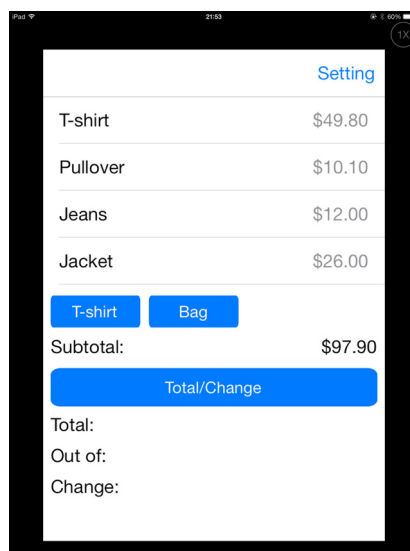
サンプルプログラムの使用方法

設定

以下の手順で使用します。

1 サンプルプログラムを起動します。詳細は、143 ページ「環境構築」を参照してください。

2 ˝ePOS-Device Sample Program for iOS˝ 画面が表示されます。[Setting] ボタンを押します。

3 以下の画面が表示されます。TM-i の IP アドレスを入力し、[Connect] ボタンを押します。

4 デバイスオブジェクトを作成します。以下の以下の手順で作成します。
1. [Printer ID] に、" local_printer" と入力し、[Create] ボタンを押します。
2. [Display ID] に、" local_display" と入力し、[Create] ボタンを押します。
3. [Scanner ID] に、" local_scanner" と入力し、[Create] ボタンを押します。

5 [Back] ボタンを押します。

操作

以下の手順で、レシートを印刷します。

1. サンプルプログラムを起動します。詳細は、環境構築（143 ページ）を参照してください。
2. サンプルプログラムの設定をします。詳細は、設定（145 ページ）を参照してください。
3. 商品を登録します。以下の方法で登録します。
登録すると、カスタマーディスプレイに、商品と金額が表示されます。
   - [T-shirt] ボタンを押す。
   - [Bag] ボタンを押す。
   - バーコードをバーコードスキャナーで読み込む。

参考	商品のバーコードは、サンプルプログラムに PDF で付属されます。印刷してお使いください。

4. [Total/Change] ボタンを押します。
カスタマーディスプレイに合計金額が表示されます。
5. [Out of:] に預かり金を入力します。
6. [Total/Change] ボタンを押します。
カスタマーディスプレイにお釣りが表示され、レシートが印刷されます。

付録

キーコード一覧

定数	コード	定数	コード	定数	コード
EDEV_OC_VK_BACK	0x08	EDEV_OC_VK_7	0x37	EDEV_OC_VK_Z	0x5A
EDEV_OC_VK_TAB	0x09	EDEV_OC_VK_8	0x38	EDEV_OC_VK_MULTIPLY	0x6A
EDEV_OC_VK_RETURN	0x0D	EDEV_OC_VK_9	0x39	EDEV_OC_VK_ADD	0x6B
EDEV_OC_VK_SHIFT	0x10	EDEV_OC_VK_A	0x41	EDEV_OC_VK_SUBTRACT	0x6D
EDEV_OC_VK_CONTROL	0x11	EDEV_OC_VK_B	0x42	EDEV_OC_VK_F1	0x70
EDEV_OC_VK_MENU	0x12	EDEV_OC_VK_C	0x43	EDEV_OC_VK_F2	0x71
EDEV_OC_VK_CAPITAL	0x14	EDEV_OC_VK_D	0x44	EDEV_OC_VK_F3	0x72
EDEV_OC_VK_ESCAPE	0x1B	EDEV_OC_VK_E	0x45	EDEV_OC_VK_F4	0x73
EDEV_OC_VK_CONVERT	0x1C	EDEV_OC_VK_F	0x46	EDEV_OC_VK_F5	0x74
EDEV_OC_VK_NONCONVERT	0x1D	EDEV_OC_VK_G	0x47	EDEV_OC_VK_F6	0x75
EDEV_OC_VK_SPACE	0x20	EDEV_OC_VK_H	0x48	EDEV_OC_VK_F7	0x76
EDEV_OC_VK_PRIOR	0x21	EDEV_OC_VK_I	0x49	EDEV_OC_VK_F8	0x77
EDEV_OC_VK_NEXT	0x22	EDEV_OC_VK_J	0x4A	EDEV_OC_VK_F9	0x78
EDEV_OC_VK_END	0x23	EDEV_OC_VK_K	0x4B	EDEV_OC_VK_F10	0x79
EDEV_OC_VK_HOME	0x24	EDEV_OC_VK_L	0x4C	EDEV_OC_VK_F11	0x7A
EDEV_OC_VK_LEFT	0x25	EDEV_OC_VK_M	0x4D	EDEV_OC_VK_F12	0x7B
EDEV_OC_VK_UP	0x26	EDEV_OC_VK_N	0x4E	EDEV_OC_VK_OEM_1	0xBA
EDEV_OC_VK_RIGHT	0x27	EDEV_OC_VK_O	0x4F	EDEV_OC_VK_OEM_PLUS	0xBB
EDEV_OC_VK_DOWN	0x28	EDEV_OC_VK_P	0x50	EDEV_OC_VK_OEM_COMMA	0xBC
EDEV_OC_VK_INSERT	0x2D	EDEV_OC_VK_Q	0x51	EDEV_OC_VK_OEM_MINUS	0xBD
EDEV_OC_VK_DELETE	0x2E	EDEV_OC_VK_R	0x52	EDEV_OC_VK_OEM_PERIOD	0xBE
EDEV_OC_VK_0	0x30	EDEV_OC_VK_S	0x53	EDEV_OC_VK_OEM_2	0xBF
EDEV_OC_VK_1	0x31	EDEV_OC_VK_T	0x54	EDEV_OC_VK_OEM_3	0xC0
EDEV_OC_VK_2	0x32	EDEV_OC_VK_U	0x55	EDEV_OC_VK_OEM_4	0xDB
EDEV_OC_VK_3	0x33	EDEV_OC_VK_V	0x56	EDEV_OC_VK_OEM_5	0xDC
EDEV_OC_VK_4	0x34	EDEV_OC_VK_W	0x57	EDEV_OC_VK_OEM_6	0xDD
EDEV_OC_VK_5	0x35	EDEV_OC_VK_X	0x58	EDEV_OC_VK_OEM_7	0xDE
EDEV_OC_VK_6	0x36	EDEV_OC_VK_Y	0x59	EDEV_OC_VK_OEM_ATTN	0xF0

プリンターの仕様

TM-T70-i

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		203 x 203 dpi	
印字幅		416 ドット	576 ドット
印字桁数	フォント A	ANK 34 桁 / 漢字 17 桁	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 52 桁 / 漢字 26 桁	ANK 72 桁 / 漢字 36 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 8 x 16 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域		416 x 1662 ドット	576 x 1662 ドット
ラスターイメージ		モノクロ画像	
ロゴ		モノクロ画像	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2 次元シンボル		QR Code	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		非サポート	
用紙レイアウト設定		非サポート	

TM-T88V-i

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		180 x 180 dpi	
印字幅		360 ドット	512 ドット
印字桁数	フォント A	ANK 30 桁 / 漢字 15 桁	ANK 42 桁 / 漢字 21 桁
	フォント B	ANK 40 桁	ANK 56 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 9 x 17 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 16 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
ラスターイメージ		モノクロ画像 , 多階調画像	
ロゴ		モノクロ画像 , 多階調画像	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composit Symbology 非サポート)	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		オプション	
用紙レイアウト設定		非サポート	

TM-L90

		レシート	ダイカットラベル紙
インターフェイス		Ethernet/ 無線 LAN	
解像度		203 x 203 dpi	
印字幅		256 ドット〜 576 ドット	224 ドット〜 560 ドット
印字桁数	フォント A	ANK 48 桁 / 漢字 24 桁	ANK 46 桁 / 漢字 23 桁
	フォント B	ANK 57 桁 / 漢字 28 桁	ANK 56 桁 / 漢字 28 桁
	フォント C	ANK 72 桁 / 漢字 36 桁	ANK 70 桁 / 漢字 35 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 10 x 24 ドット / 漢字 20 x 24 ドット	
	フォント C	ANK 8 x 16 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 21 ドット目	
	フォント C	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色 第 1 色 , 第 2 色 (2 色印字設定時)	
ページモード初期領域		576 x 738 ドット	560 x 738 ドット
	2 色印字	576 x 369 ドット	560 x 369 ドット
ページモード最大領域		576 x 1476 ドット	560 x 1476 ドット
	2 色印字	576 x 738 ドット	560 x 738 ドット
ラスターイメージ		モノクロ画像 , 2 色画像 (2 色印字するには、メモリースイッチ設定ユーティリティーでプリンターの設定を変更してください。)	
ロゴ		モノクロ画像 , 2 色画像 (2 色印字するには、メモリースイッチ設定ユーティリティーでプリンターの設定を変更してください。)	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2 次元シンボル		PDF417, QR Code、MaxiCode	
罫線		非サポート	
用紙のカット		カット / フィードカット	

	レシート	ダイカットラベル紙
ドロアーキック	サポート	
ブザー	ドロアーキックによる	
用紙レイアウト設定	サポート（自動設定モードあり）	

TM-P20

		58 mm 仕様
インターフェイス		無線 LAN
解像度		203 x 203 dpi
印字幅		384 ドット
印字桁数	フォント A	ANK 32 桁 / 漢字 16 桁
	フォント B	ANK 42 桁 / 漢字 19 桁
	フォント C	ANK 42 桁 / 漢字 24 桁
	フォント D	ANK 38 桁
	フォント E	ANK 48 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット
	フォント B	ANK 9 x 24 ドット / 漢字 20 x 24 ドット
	フォント C	ANK 9 x 17 ドット / 漢字 16 x 16 ドット
	フォント D	ANK 10 x 24 ドット
	フォント E	ANK 8 x 16 ドット
文字のベースライン	フォント A	文字の上端から 21 ドット目
	フォント B	文字の上端から 21 ドット目
	フォント C	文字の上端から 16 ドット目（漢字は 15 ドット目）
	フォント D	文字の上端から 21 ドット目
	フォント E	文字の上端から 15 ドット目
初期改行量		30 ドット
色指定		第 1 色
ページモード初期領域		384 x 2400 ドット
ページモード最大領域		384 x 2400 ドット
ラスターイメージ		モノクロ画像
ロゴ		モノクロ画像
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded

	58 mm 仕様
2 次元シンボル	PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, DataMatrix (Composite Symbology 非サポート)
罫線	非サポート
用紙のカット	カット位置への紙送り
ドロアーキック	非サポート
ブザー	サポート (パターン 1 〜パターン 10, 停止)
用紙レイアウトの設定	サポート

TM-P60II

		レシート 58 mm	レシート 60 mm	ダイカットラベル
インターフェイス		無線 LAN		
解像度		203 x 203 dpi		
印字幅		420 ドット	432 ドット	160 (29.5 mm) ～ 400 ドット (60 mm)
印字桁数	フォント A	ANK 35 桁 / 漢字 17 桁	ANK 36 桁 / 漢字 18 桁	ANK 33 桁 / 漢字 16 桁
	フォント B	ANK 42 桁	ANK 43 桁	ANK 40 桁
	フォント C	ANK 52 桁	ANK 54 桁	ANK 50 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット		
	フォント B	ANK 10 x 24 ドット		
	フォント C	ANK 8 x 16 ドット		
文字のベースライン	フォント A	文字の上端から 21 ドット目		
	フォント B	文字の上端から 16 ドット目		
	フォント C	文字の上端から 15 ドット目		
初期改行量		30 ドット		
色指定		第 1 色		
ページモード初期領域		420 x 1624 ドット	432 x 1624 ドット	400 x 1624 ドット
ページモード最大領域		420 x 1624 ドット	432 x 1624 ドット	400 x 1624 ドット
ラスターイメージ		モノクロ画像		
ロゴ		モノクロ画像		
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded		
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked Aztec Code, DataMatrix (Composit Symbology 非サポート)		
罫線		非サポート		
用紙のカット		カット / フィードカット		
ドロアーキック		サポート		
ブザー		サポート (パターン 1 ～ 10, 停止)		
用紙レイアウト設定		サポート		

TM-T70

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		203 x 203 dpi	
印字幅		416 ドット	576 ドット
印字桁数	フォント A	ANK 34 桁 / 漢字 17 桁	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 52 桁 / 漢字 26 桁	ANK 72 桁 / 漢字 36 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 8 x 16 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域		416 x 1662 ドット	576 x 1662 ドット
ラスターイメージ		モノクロ画像	
ロゴ		モノクロ画像	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2 次元シンボル		QR Code	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		非サポート	
用紙レイアウト設定		非サポート	

TM-T70II

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		203 x 203 dpi	
印字幅		416 ドット	576 ドット
印字桁数	フォント A	ANK 34 桁 / 漢字 17 桁	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 52 桁 / 漢字 26 桁	ANK 72 桁 / 漢字 36 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 9 x 17 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域		416 x 1662 ドット	576 x 1662 ドット
ラスターイメージ		モノクロ画像	
ロゴ		モノクロ画像	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		サポート	
用紙レイアウト設定		非サポート	

TM-T88IV

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		180 x 180 dpi	
印字幅		360 ドット	512 ドット
印字桁数	フォント A	ANK 30 桁 / 漢字 15 桁	ANK 42 桁 / 漢字 21 桁
	フォント B	ANK 40 桁	ANK 56 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 9 x 17 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 16 ドット目	
初期改行量		30 ドット	
色指定		第 1 色、第 2 色 (2 色印字設定時)	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
	2 色印字	360 x 415 ドット	512 x 415 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
	2 色印字	360 x 831 ドット	512 x 831 ドット
ラスターイメージ		モノクロ画像 , 2 色画像	
ロゴ		モノクロ画像 , 2 色画像 (2 色印字するには、メモリースイッチ設定ユーティリティーでプリンターの設定を変更してくださ い。)	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2 次元シンボル		QR Code	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		非サポート	
用紙レイアウト設定		非サポート	

TM-T88V

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		180 x 180 dpi	
印字幅		360 ドット	512 ドット
印字桁数	フォント A	ANK 30 桁 / 漢字 15 桁	ANK 42 桁 / 漢字 21 桁
	フォント B	ANK 40 桁	ANK 56 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 9 x 17 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 16 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
ラスターイメージ		モノクロ画像 , 多階調画像	
ロゴ		モノクロ画像 , 多階調画像	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composit Symbology 非サポート)	
罫線		非サポート	
用紙のカット		カット / フィード カット	
ドロアーキック		サポート	
ブザー		オプション	
用紙レイアウト設定		非サポート	

TM-T90

		58 mm 仕様	80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN	
解像度		203 x 203 dpi	
印字幅		420 ドット	576 ドット
印字桁数	フォント A	ANK 35 桁 / 漢字 17 桁	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 42 桁 / 漢字 21 桁	ANK 57 桁 / 漢字 28 桁
	フォント C	ANK 52 桁 / 漢字 26 桁	ANK 72 桁 / 漢字 36 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 10 x 24 ドット / 漢字 20 x 24 ドット	
	フォント C	ANK 8 x 16 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 21 ドット目	
	フォント C	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色 第 1 色 , 第 2 色 (2 色印字設定時)	
ページモード初期領域		420 x 738 ドット	576 x 738 ドット
	2 色印字	420 x 369 ドット	576 x 369 ドット
ページモード最大領域		420 x 1476 ドット	576 x 1476 ドット
	2 色印字	420 x 738 ドット	576 x 738 ドット
ラスターイメージ		モノクロ画像 , 2 色画像 (2 色印字するには、メモリースイッチ設定ユーティリティーでプリンターの設定を変更してください。)	
ロゴ		モノクロ画像 , 2 色画像 (2 色印字するには、メモリースイッチ設定ユーティリティーでプリンターの設定を変更してください。)	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2 次元シンボル		PDF417	
罫線		非サポート	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	

	58 mm 仕様	80 mm 仕様
ブザー	ドロアーキックによる	
用紙レイアウト設定	非サポート	

TM-T90KP

		80 mm 仕様
インターフェイス		Ethernet/ 無線 LAN
解像度		203 x 203 dpi
印字幅		576 ドット
印字桁数	フォント A	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 57 桁 / 漢字 28 桁
	フォント C	ANK 72 桁 / 漢字 36 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット
	フォント B	ANK 10 x 24 ドット / 漢字 20 x 24 ドット
	フォント C	ANK 8 x 16 ドット / 漢字 16 x 16 ドット
文字のベースライン	フォント A	文字の上端から 21 ドット目
	フォント B	文字の上端から 21 ドット目
	フォント C	文字の上端から 15 ドット目
初期改行量		30 ドット
色指定		第 1 色
ページモード初期領域		576 x 738 ドット
ページモード最大領域		576 x 1476 ドット
ラスターイメージ		モノクロ画像
ロゴ		モノクロ画像
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128
2 次元シンボル		非サポート
罫線		サポート
用紙のカット		カット / フィードカット / カット予約
ドロアーキック		非サポート
ブザー		サポート
用紙レイアウト設定		非サポート

TM-DT シリーズで使用する場合

ePOS-Device SDK for iOS は、TM-DT シリーズ (TM-T88V-DT/TM-T70II-DT) でも使用できます。
ただし、TM-DT シリーズでは、以下の機能を使用できません。

❏ 本書で「TM-i ファームウェア Ver.4.1 以降で対応」と記載されている機能
- アプリケーションから印刷ジョブ ID を指定 (12 ページ)
- スプーラー機能 (13 ページ)

❏ TM-i シリーズにない TM-DT シリーズ特有の機能
- デバイス制御スクリプトを使用した各種 POS 周辺機器の制御
(詳細は、TM-DT シリーズの「詳細取扱説明書」を参照してください。)

上記のほかに、ePOS-Device SDK for iOS で用意している API には、いくつかの制限事項があります。詳細は、ePOS-Device SDK for iOS のパッケージに含まれている「README.jp.txt」を参照してください。